Missão Prática | Nível 1 | Mundo 3

RPG0014 - Iniciando o caminho pelo Java.

Aluna: Simone Ramos de Jesus.

Matricula: 202208290965.

Campus: Polo Prado – Belo Horizonte – MG.

Curso: Desenvolvimento Full Stack

Turma: 2023.3

GITHUB:

Objetivos da prática:

1. Utilizar herança e polimorfismo na definição de entidades.
2. Utilizar persistência de objetos em arquivos binários.
3. Implementar uma interface cadastral em modo texto.
4. Utilizar o controle de exceções da plataforma Java.
7. persistência em arquivos binários.

Model-Entidades-Pessoa:

Pessoa.java

model > entidades > Pessoa.java

```java
package model.entidades;

import java.io.Serializable;

public abstract class Pessoa implements Serializable {
    private String nome;
    private int id;

    public Pessoa() {
    }

    public Pessoa(String nome, int id) {
        this.nome = nome;
        this.id = id;
    }

    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() throws Exception {
        System.out.println("Nome: " + nome);
        System.out.println("ID: " + id);
    }
}
```

Model- Entidades- Pessoa Fisíca:

PessoaFisica.java

model > entidades > PessoaFisica.java

```java
package model.entidades;

import java.io.Serializable;
import java.time.LocalDate;

public class PessoaFisica extends Pessoa implements Serializable {
  private String cpf;
  private int idade;

  public PessoaFisica() {
  }

  public PessoaFisica(String nome, String cpf, int idade, int id) {
    super(nome, id);
    this.idade = idade;
    this.cpf = cpf;
  }

  public PessoaFisica(String string, String string2, LocalDate localDate) {
  }

  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() {
    System.out.println("***Pessoa Física***");
    System.out.println("ID: " + getId());
    System.out.println("Nome: " + getNome());
    System.out.println("CPF: " + getCpf());
    System.out.println("Idade: " + getIdade());
  }
}
```

Model – Entidades – Pessoa Jurídica:

PessoaJuridica.java

model > entidades > PessoaJuridica.java

```java
package model.entidades;

import java.io.Serializable;

public class PessoaJuridica extends Pessoa implements Serializable {
  private String cnpj;

  public PessoaJuridica() {
  }

  public PessoaJuridica(String nome, String cnpj, int id) {
    super(nome, id);
    this.cnpj = cnpj;
  }

  public PessoaJuridica(String nomeCompleto, int id, Object cnpj2) {
  }

  public String getCnpj() {
    return cnpj;
  }

  public void setCnpj(String cnpj) {
    this.cnpj = cnpj;
  }

  @Override
  public void exibir() {
    System.out.println("***Pessoa Jurídica***");
    System.out.println("ID: " + getId());
    System.out.println("Nome: " + getNome());
    System.out.println("CNPJ: " + getCnpj());
  }

}

```

Model – Gerenciador – Pessoa Física Repo:

PessoaFisicaRepo.java

model > gerenciador > PessoaFisicaRepo.java

```java
package model.gerenciador;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import model.entidades.PessoaFisica;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;

public class PessoaFisicaRepo {
  private List<PessoaFisicaRepo> pessoasFisicasLista = new ArrayList<>();
  public PessoaFisicaRepo pessoasFisicas;

  public void inserir(PessoaFisica pf) {
    getPessoasFisicasLista().add(pf);
  }

  public List<PessoaFisicaRepo> getPessoasFisicasLista() {
    return pessoasFisicasLista;
  }

  public void setPessoasFisicasLista(List<PessoaFisicaRepo> list) {
    this.pessoasFisicasLista = list;
  }

  public void alterar(PessoaFisicaRepo pessoaFisicaExcluir) {
    int index = getPessoasFisicasLista().indexOf(pessoaFisicaExcluir);
    if (index != -1) {
      getPessoasFisicasLista().set(index, pessoaFisicaExcluir);
    }
  }
```

PessoaFisicaRepo.java

model > gerenciador > PessoaFisicaRepo.java

```java
    /**
     * @param id
     */
    public void excluir(int id) {
        getPessoasFisicasLista().removeIf(pessoa -> pessoa.getId() == id);
    }

    int getId() {
        return 0;
    }

    public PessoaFisicaRepo obter(int id) {
        for (PessoaFisicaRepo pessoa : getPessoasFisicasLista()) {
            if (pessoa.getId() == id) {
                return pessoa;
            }
        }
        return null;
    }

    public List<PessoaFisicaRepo> obterTodos() {
        return this.getPessoasFisicasLista();
    }

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

PessoaFisicaRepo.java

model > gerenciador > PessoaFisicaRepo.java

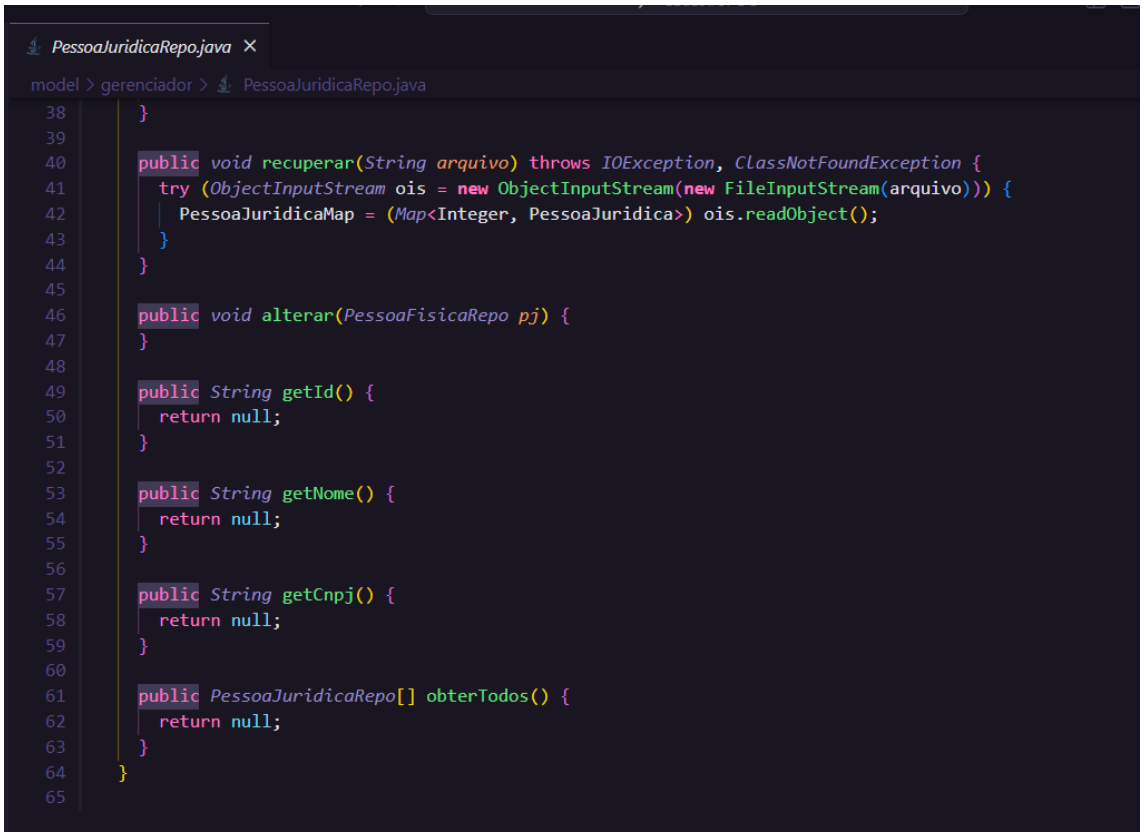
```java
    /**
     * @param nameArquivo
     * @throws IOException
     * @throws ClassNotFoundException
     */
    public void recuperar(String nameArquivo) throws IOException, ClassNotFoundException {
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(nameArquivo))) {
            setPessoasFisicasLista((List<PessoaFisicaRepo>) ois.readObject());
        }
    }
    public void setIdade(int parseInt) {}
    public void setCpf(String readLine) {}
    public String getIdade() {
        return null;}
    public String getNome() {
        return null; }
    public void setNome(String readLine) {}
    public String getCnpj() {
        return null;}
    public void setCnpj(String readLine) {}
    public String getNomeEmpresa() {
        return null;}
    public void setNomeEmpresa(String readLine) {}
    public PessoaFisica[] obterTodos(int i) {
        return null;}
    public String getCpf() {
        return null;}
}
```

Model – Gerenciador – Pessoa Jurídica Repo:

PessoaJuridicaRepo.java

model > gerenciador > PessoaJuridicaRepo.java

```java
package model.gerenciador;

import java.io.*;
import java.util.HashMap;
import java.util.Map;
import java.util.List;

import model.entidades.PessoaFisica;
import model.entidades.PessoaJuridica;

public class PessoaJuridicaRepo {
  private Map<Integer, PessoaJuridica> PessoaJuridicaMap;

  public PessoaJuridicaRepo() {
    PessoaJuridicaMap = new HashMap<>();
  }

  public void inserir(PessoaJuridica pj) {
    PessoaJuridicaMap.put(pj.getId(), pj);
  }

  public void alterar(PessoaJuridica pessoaJuridica) {
    this.PessoaJuridicaMap.put(pessoaJuridica.getId(), pessoaJuridica);
  }

  public void excluir(int id) {
    PessoaJuridicaMap.remove(id);
  }

  public PessoaJuridica obter(int id) {
    return PessoaJuridicaMap.get(id);
  }

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

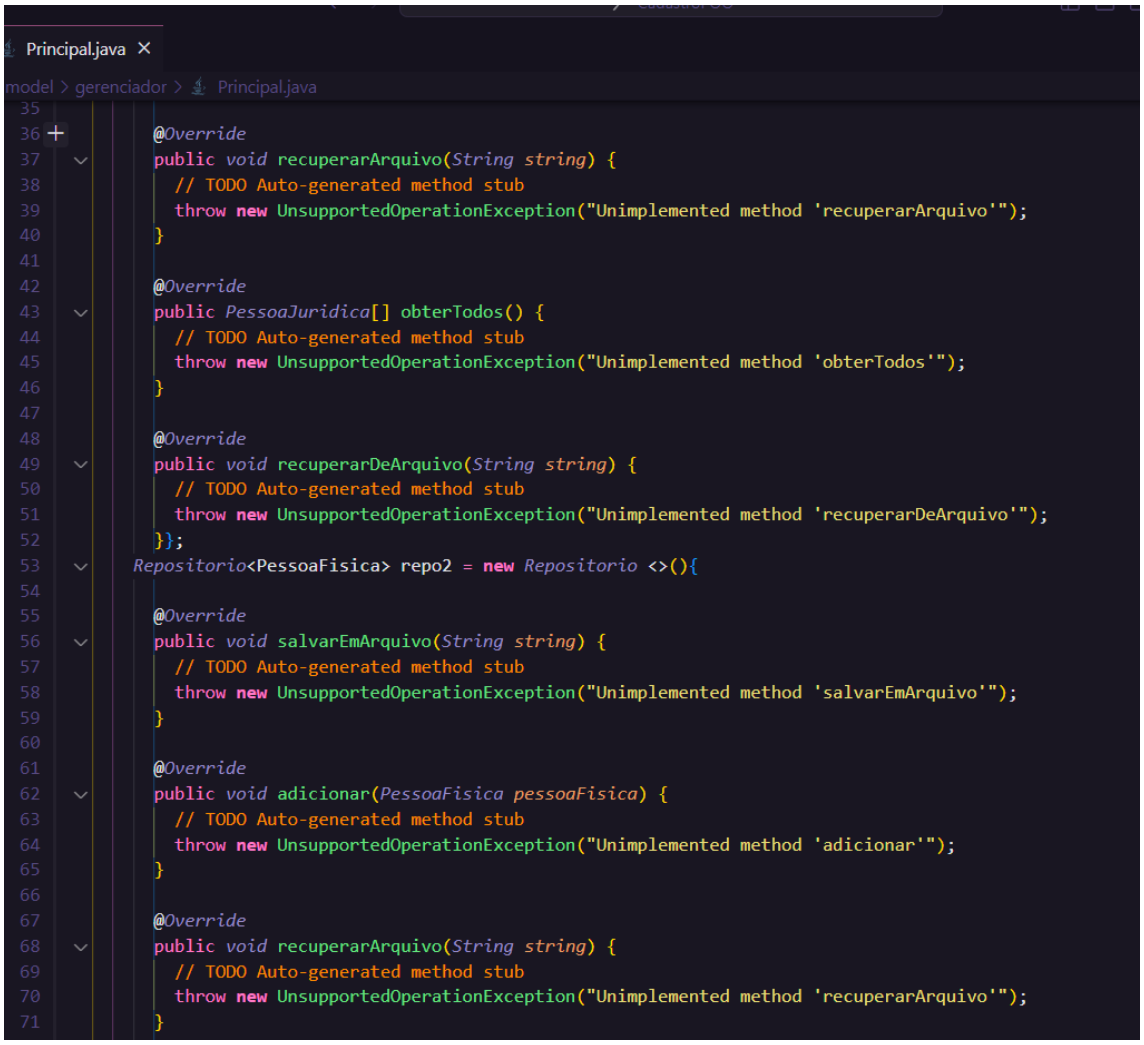

PessoaJuridicaRepo.java
model > gerenciador > PessoaJuridicaRepo.java
38 }
39
40 public void recuperar(String arquivo) throws IOException, ClassNotFoundException {
41 try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(arquivo))) {
42 PessoaJuridicaMap = (Map<Integer, PessoaJuridica>) ois.readObject();
43 }
44 }
45
46 public void alterar(PessoaFisicaRepo pj) {
47 }
48
49 public String getId() {
50 return null;
51 }
52
53 public String getNome() {
54 return null;
55 }
56
57 public String getCnpj() {
58 return null;
59 }
60
61 public PessoaJuridicaRepo[] obterTodos() {
62 return null;
63 }
64 }
65

Model – Gerenciador – Principal:

Principal.java

model > gerenciador > Principal.java

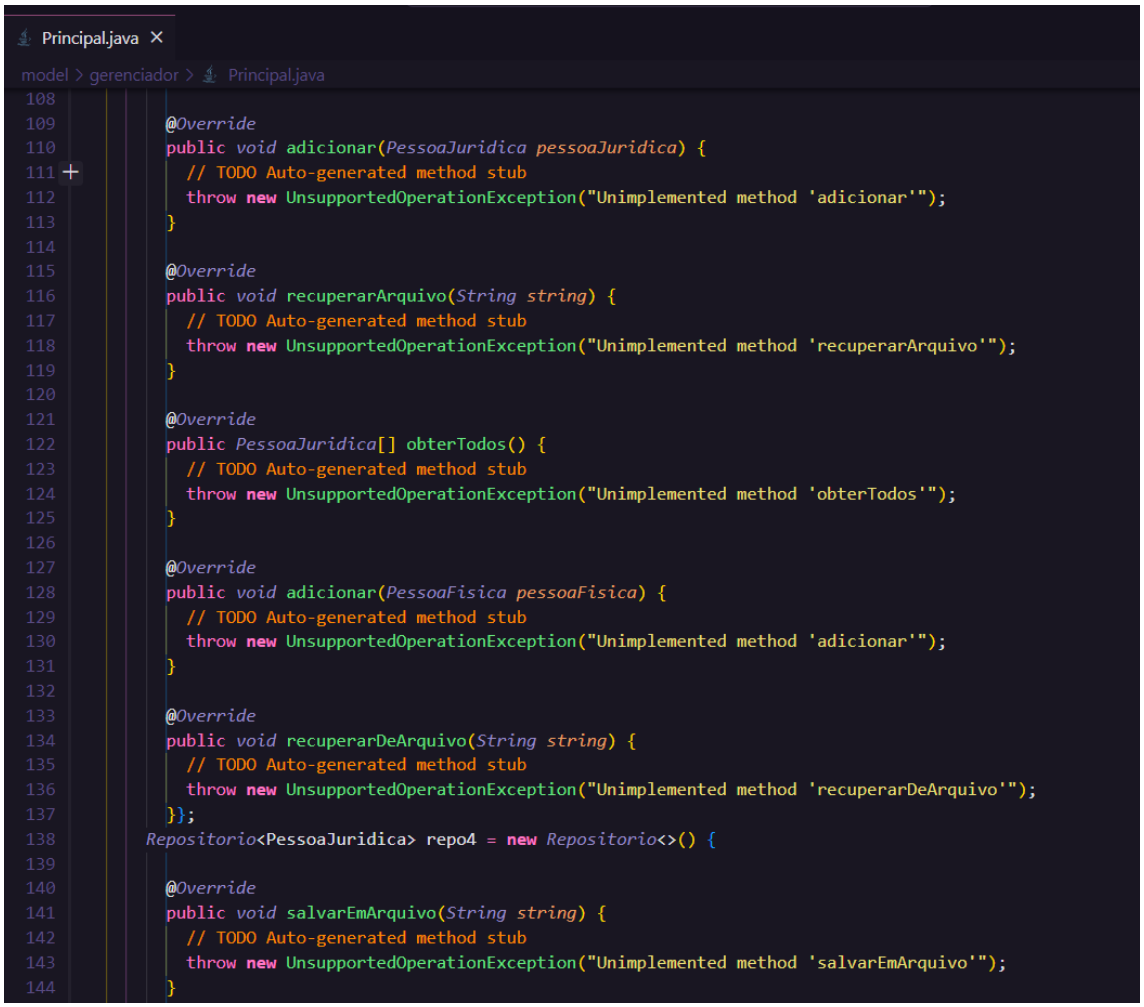
```java
package model.gerenciador;

import java.io.IOException;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;

import System.out.Repositorio;
import model.entidades.Pessoa;
import model.entidades.PessoaFisica;
import model.entidades.PessoaJuridica;
import model.gerenciador.PessoaFisicaRepo;
import model.gerenciador.PessoaJuridicaRepo;
import model.gerenciador.Principal;

public class Principal{

  private static final String Repositorio = null;

  public static void main(String[] args) throws IOException{

    Repositorio<PessoaFisica> repo1 = new Repositorio <>(){

      @Override
      public void salvarEmArquivo(String string) {
        // TODO Auto-generated method stub
        throw new UnsupportedOperationException("Unimplemented method 'salvarEmArquivo'");
      }

      @Override
      public void adicionar(PessoaFisica pessoaFisica) {
        // TODO Auto-generated method stub
        throw new UnsupportedOperationException("Unimplemented method 'adicionar'");
      }

```


Principal.java
model > gerenciador > Principal.java
@Override
public void recuperarArquivo(String string) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'recuperarArquivo'");
}
@Override
public PessoaJuridica[] obterTodos() {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'obterTodos'");
}
@Override
public void recuperarDeArquivo(String string) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'recuperarDeArquivo'");
}};
Repositorio<PessoaFisica> repo2 = new Repositorio <>(){
@Override
public void salvarEmArquivo(String string) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'salvarEmArquivo'");
}
@Override
public void adicionar(PessoaFisica pessoaFisica) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'adicionar'");
}
@Override
public void recuperarArquivo(String string) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'recuperarArquivo'");
}

Principal.java

odel > gerenciador > Principal.java

```java
        @Override
        public PessoaJuridica[] obterTodos() {
          // TODO Auto-generated method stub
          throw new UnsupportedOperationException("Unimplemented method 'obterTodos'");
        }

        @Override
        public void recuperarDeArquivo(String string) {
          // TODO Auto-generated method stub
          throw new UnsupportedOperationException("Unimplemented method 'recuperarDeArquivo'");
        }};

      repo1.adicionar(new PessoaFisica("Simone", "01234", LocalDate.of(1985, 11, 18)));
      repo1.adicionar(new PessoaFisica("Paula", "98765", LocalDate.of(1987, 04, 15)));

      repo1.salvarEmArquivo("teste.fisica.bin");

      try{
        repo2.recuperarDeArquivo("teste.fisica.bin");
      }
      catch (IOException|ClassNotFoundException e){
        e.printStackTrace();
      }

      for (PessoaFisicaRepo pf : ((PessoaFisicaRepo) repo2).obterTodos()){
        System.out.println(pf);
      }

      Repositorio<PessoaJuridica> repo3 = new Repositorio<>() {

        @Override
        public void salvarEmArquivo(String string) {
          // TODO Auto-generated method stub
          throw new UnsupportedOperationException("Unimplemented method 'salvarEmArquivo'");
        }
```


Principal.java
model > gerenciador > Principal.java
@Override
public void adicionar(PessoaJuridica pessoaJuridica) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'adicionar'");
}
@Override
public void recuperarArquivo(String string) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'recuperarArquivo'");
}
@Override
public PessoaJuridica[] obterTodos() {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'obterTodos'");
}
@Override
public void adicionar(PessoaFisica pessoaFisica) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'adicionar'");
}
@Override
public void recuperarDeArquivo(String string) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'recuperarDeArquivo'");
}};
Repositorio<PessoaJuridica> repo4 = new Repositorio<>() {
@Override
public void salvarEmArquivo(String string) {
// TODO Auto-generated method stub
throw new UnsupportedOperationException("Unimplemented method 'salvarEmArquivo'");
}

Principal.java

model > gerenciador > Principal.java

```java
        @Override
        public void adicionar(PessoaFisica pessoaFisica) {
          // TODO Auto-generated method stub
          throw new UnsupportedOperationException("Unimplemented method 'adicionar'");
        }

        @Override
        public void recuperarArquivo(String string) {
          // TODO Auto-generated method stub
          throw new UnsupportedOperationException("Unimplemented method 'recuperarArquivo'");
        }

        @Override
        public PessoaJuridica[] obterTodos() {
          // TODO Auto-generated method stub
          throw new UnsupportedOperationException("Unimplemented method 'obterTodos'");
        }

        @Override
        public void recuperarDeArquivo(String string) {
          // TODO Auto-generated method stub
          throw new UnsupportedOperationException("Unimplemented method 'recuperarDeArquivo'");
        }};

      repo3.adicionar(new PessoaJuridica("Empresa On", 9870, LocalDate.of(0, 0, 0)));
      repo4.adicionar(new PessoaJuridica("Empresa Off", 5678, LocalDate.of(0, 0, 0)));

      repo3.salvarEmArquivo("teste.juridica.bin");

      repo4.recuperarArquivo("teste.juridica.bin");

      for(PessoaJuridica pj : repo4.obterTodos()){
        System.out.println(pj);
      }
    }
  }
```

Model – Gerenciador – Main:

main.java

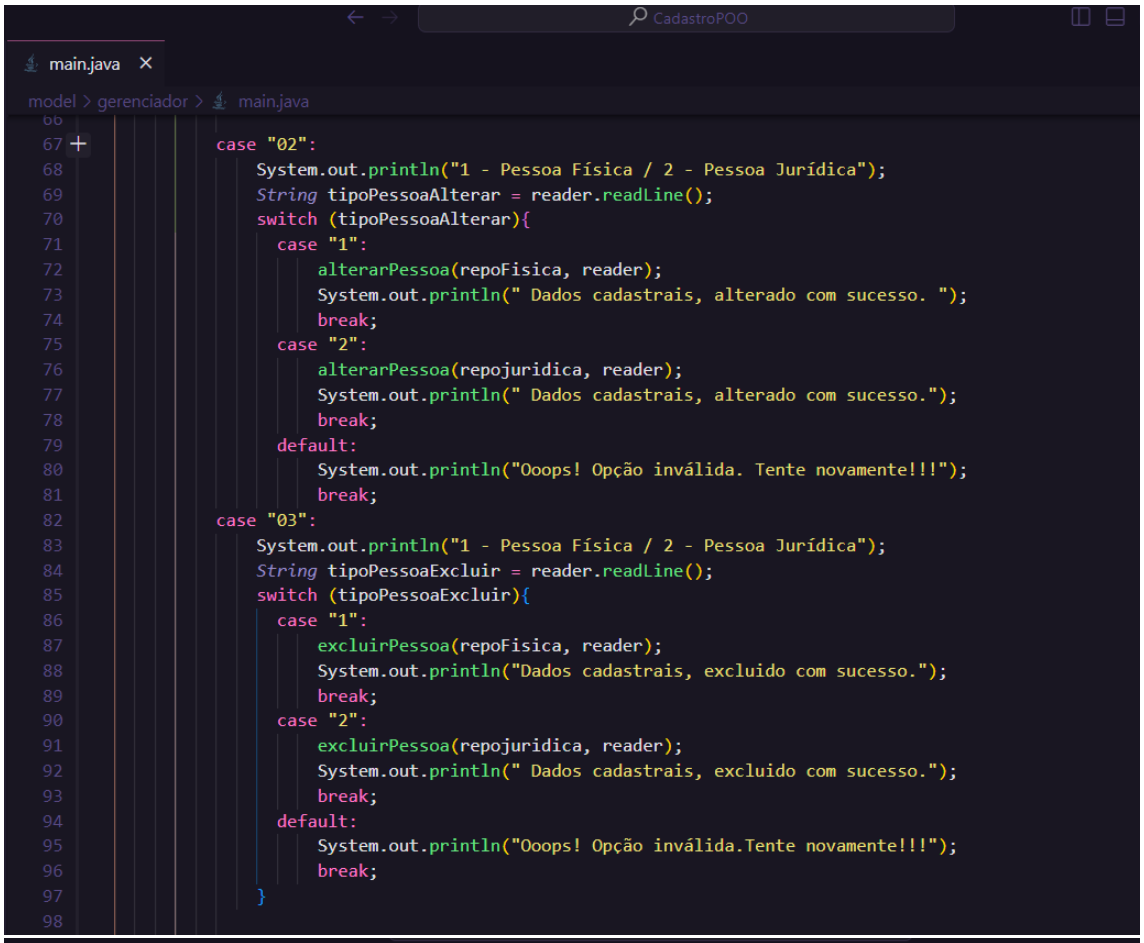
model > gerenciador > main.java

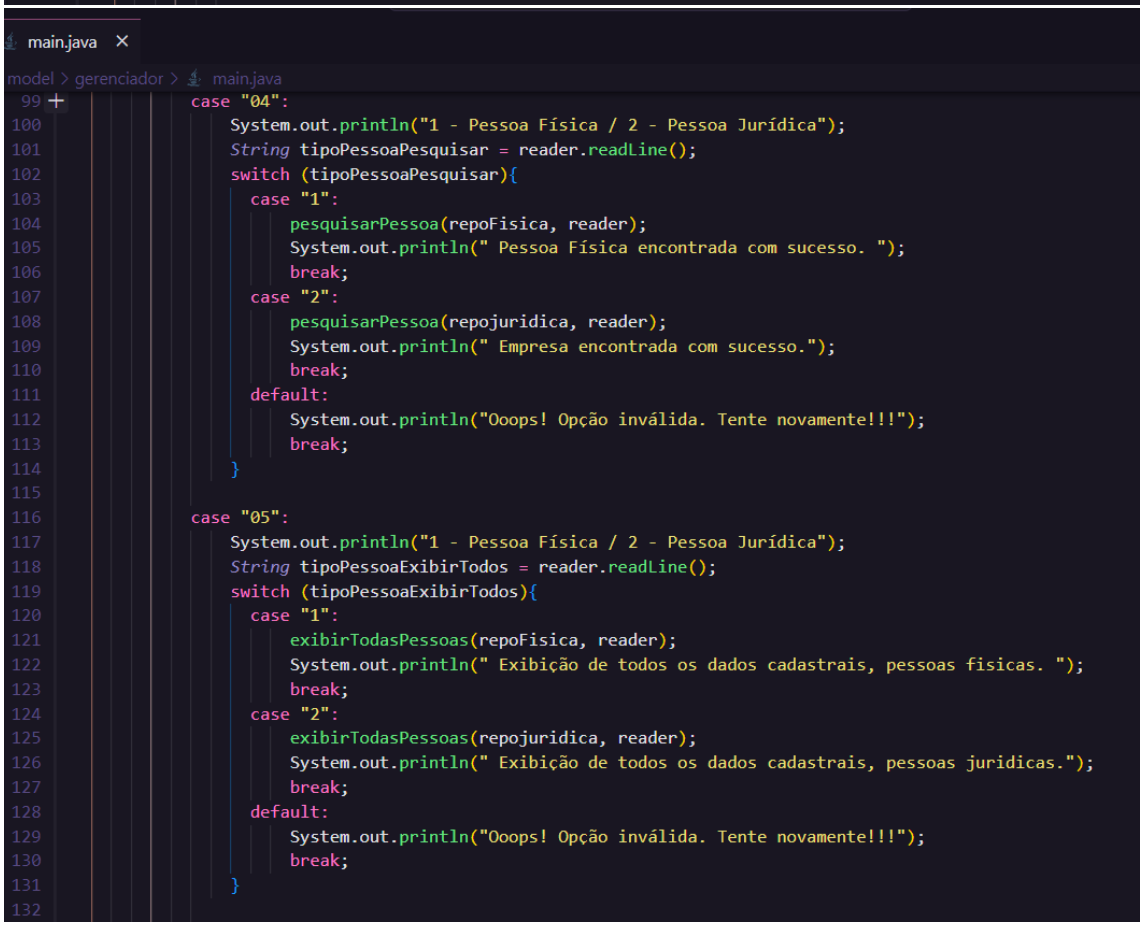
```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;

import model.entidades.Pessoa;
import model.entidades.PessoaFisica;
import model.entidades.PessoaJuridica;
import model.gerenciador.PessoaFisicaRepo;
import model.gerenciador.PessoaJuridicaRepo;
import model.gerenciador.Principal;

public class main {
  private static BufferedReader reader;
  private static String nome;
  private static Object cnpj;

  /**
   * @param args
   */
public static void main(String[] args){
    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
    String opcao = " ";
    PessoaFisicaRepo repoFisica = new PessoaFisicaRepo();
    PessoaJuridicaRepo repojuridica = new PessoaJuridicaRepo();
```

main.java

model > gerenciador > main.java

```java

        while ( !"x".equalsIgnoreCase(opcao)){
          System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");
          System.out.println("Selecione a opção desejada: ");
          System.out.println("01 - Inserir Pessoa");
          System.out.println("02 - Alterar Pessoa");
          System.out.println("03 - Excluir Pessoa");
          System.out.println("04 - Pesquisar Por ID");
          System.out.println("05 - Exibir Todos");
          System.out.println("06 - Persistir Dados");
          System.out.println("07 - Recuperar Dados");
          System.out.println("00 - Exit");
          System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");
        }

          try{
            opcao = reader.readLine();
            switch (opcao){
              case "01":
                  System.out.println("1 - Pessoa Física / 2 - Pessoa Jurídica");
                  String tipoPessoa = reader.readLine();
                  switch (tipoPessoa){
                    case "1":
                        PessoaFisica pf = lerDadosPessoaFisica(reader);
                        repoFisica.inserir(pf);
                        System.out.println(" Pessoa Fisica inserida com sucesso. ");
                        break;
                    case "2":
                        PessoaJuridica pj = lerDadosPessoaJuridica(reader);
                        repojuridica.inserir(pj);
                        System.out.println(" Pessoa Jurídica inserida com sucesso. ");
                        break;
                    default:
                        System.out.println("Ooops! Opcão inválida.Tente novamente");
                        break;
                    }
```

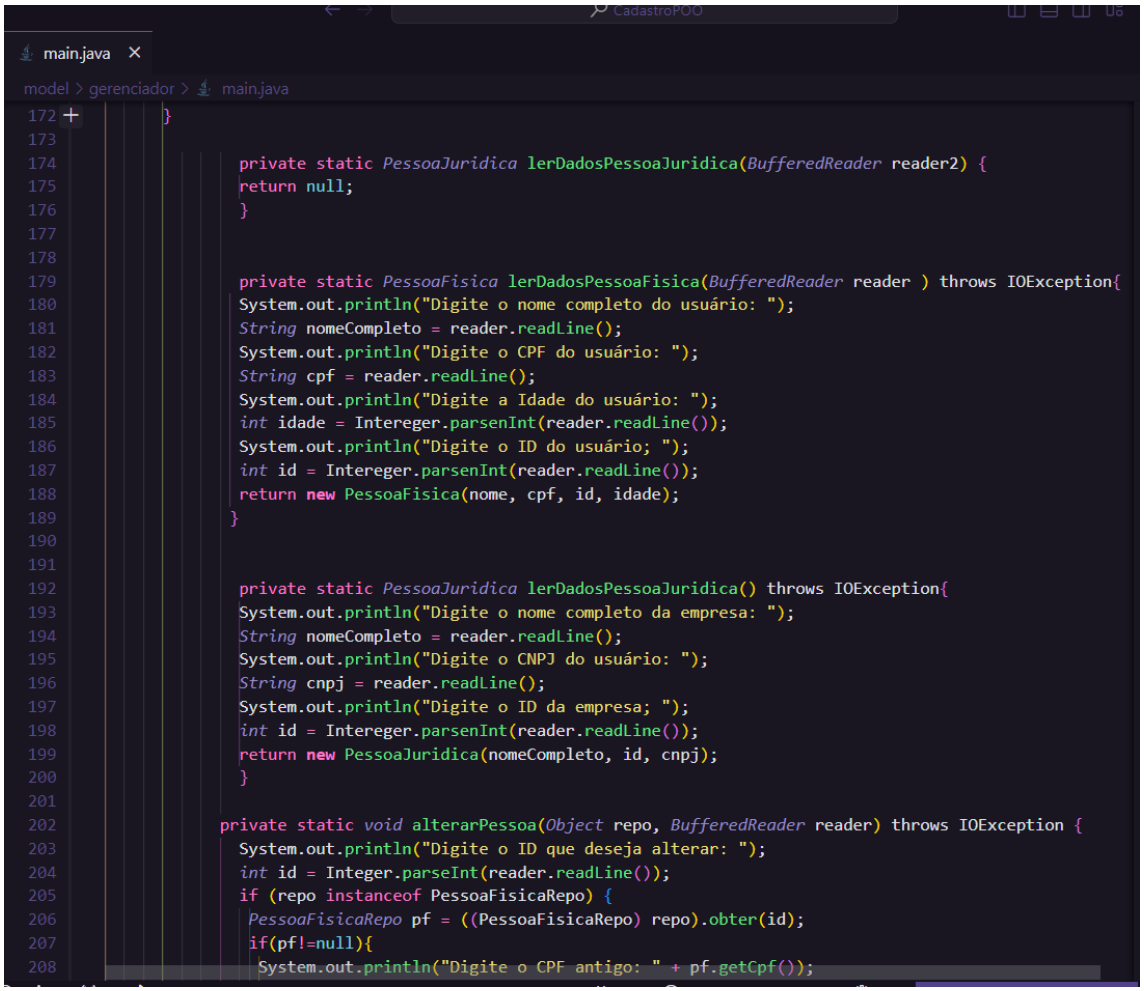

CadastroPOO
main.java
model > gerenciador > main.java
case "02":
System.out.println("1 - Pessoa Física / 2 - Pessoa Jurídica");
String tipoPessoaAlterar = reader.readLine();
switch (tipoPessoaAlterar){
case "1":
alterarPessoa(repoFisica, reader);
System.out.println(" Dados cadastrais, alterado com sucesso. ");
break;
case "2":
alterarPessoa(repojuridica, reader);
System.out.println(" Dados cadastrais, alterado com sucesso.");
break;
default:
System.out.println("Ooops! Opção inválida. Tente novamente!!!");
break;
case "03":
System.out.println("1 - Pessoa Física / 2 - Pessoa Jurídica");
String tipoPessoaExcluir = reader.readLine();
switch (tipoPessoaExcluir){
case "1":
excluirPessoa(repoFisica, reader);
System.out.println("Dados cadastrais, excluido com sucesso.");
break;
case "2":
excluirPessoa(repojuridica, reader);
System.out.println(" Dados cadastrais, excluido com sucesso.");
break;
default:
System.out.println("Ooops! Opção inválida.Tente novamente!!!");
break;
}


main.java
model > gerenciador > main.java
case "04":
System.out.println("1 - Pessoa Física / 2 - Pessoa Jurídica");
String tipoPessoaPesquisar = reader.readLine();
switch (tipoPessoaPesquisar){
case "1":
pesquisarPessoa(repoFisica, reader);
System.out.println(" Pessoa Física encontrada com sucesso. ");
break;
case "2":
pesquisarPessoa(repojuridica, reader);
System.out.println(" Empresa encontrada com sucesso.");
break;
default:
System.out.println("Ooops! Opção inválida. Tente novamente!!!");
break;
}
case "05":
System.out.println("1 - Pessoa Física / 2 - Pessoa Jurídica");
String tipoPessoaExibirTodos = reader.readLine();
switch (tipoPessoaExibirTodos){
case "1":
exibirTodasPessoas(repoFisica, reader);
System.out.println(" Exibição de todos os dados cadastrais, pessoas fisicas. ");
break;
case "2":
exibirTodasPessoas(repojuridica, reader);
System.out.println(" Exibição de todos os dados cadastrais, pessoas juridicas.");
break;
default:
System.out.println("Ooops! Opção inválida. Tente novamente!!!");
break;
}

main.java

model > gerenciador > main.java

```
133             case "06":
134                 System.out.println("Informe o prefixo dos arquivos");
135                 String arquivoX = reader.readLine();
136                 try {
137                   repoFisica.persistir(arquivoX + ".fisica.bin");
138                   repoJuridica.persistir(arquivoX + ".juridica.bin");
139                   System.out.println("Dados cadastrais, salvos com sucesso.");
140               }
141               catch(IOException e){
142                   System.out.println( "[x] Erro ao salvar os dados cadastrais: "+ e.getMessage());
143                   break;
144               }
145             case "07":
146                 System.out.println("Informe o prefixo dos arquivos: ");
147                 String arquivoG = reader.readLine();
148                 try {
149                   repoFisica.recuperar(arquivoG + ".fisica.bin");
150                   repoJuridica.recuperar(arquivoG + ".juridica.bin");
151                   System.out.println("Dados cadastrais, recuperados com sucesso!");
152                   }
153                 catch(IOException|ClassNotFoundException e) {
154                   System.out.println( "[x] Erro ao recuperar os dados cadastrais: "+ e.getMessage());
155                   break;
156                 }
157
158             case "00":
159                   System.out.println("Fechando o sistema, gentileza aguarde! ");
160                   break;
161
162                 default:
163                     System.out.println("Ooops! Opção inválida. Tente novamente!!!");
164                     break;
165
166                 catch (IOException e){
167                 final Throwable eThrowable;
168                 System.out.println( "[x] Erro ao recuperar os dados cadastrais: "+ e.getMessage());
169                 }
```


main.java
model > gerenciador > main.java
}
private static PessoaJuridica lerDadosPessoaJuridica(BufferedReader reader2) {
return null;
}
private static PessoaFisica lerDadosPessoaFisica(BufferedReader reader) throws IOException{
System.out.println("Digite o nome completo do usuário: ");
String nomeCompleto = reader.readLine();
System.out.println("Digite o CPF do usuário: ");
String cpf = reader.readLine();
System.out.println("Digite a Idade do usuário: ");
int idade = Intereger.parsenInt(reader.readLine());
System.out.println("Digite o ID do usuário; ");
int id = Intereger.parsenInt(reader.readLine());
return new PessoaFisica(nome, cpf, id, idade);
}
private static PessoaJuridica lerDadosPessoaJuridica() throws IOException{
System.out.println("Digite o nome completo da empresa: ");
String nomeCompleto = reader.readLine();
System.out.println("Digite o CNPJ do usuário: ");
String cnpj = reader.readLine();
System.out.println("Digite o ID da empresa; ");
int id = Intereger.parsenInt(reader.readLine());
return new PessoaJuridica(nomeCompleto, id, cnpj);
}
private static void alterarPessoa(Object repo, BufferedReader reader) throws IOException {
System.out.println("Digite o ID que deseja alterar: ");
int id = Integer.parseInt(reader.readLine());
if (repo instanceof PessoaFisicaRepo) {
PessoaFisicaRepo pf = ((PessoaFisicaRepo) repo).obter(id);
if(pf!=null){
System.out.println("Digite o CPF antigo: " + pf.getCpf());

main.java

model > gerenciador > main.java

```java
private static void alterarPessoa(Object repo, BufferedReader reader) throws IOException {
  System.out.println("Digite o ID que deseja alterar: ");
  int id = Integer.parseInt(reader.readLine());
  if (repo instanceof PessoaFisicaRepo) {
   PessoaFisicaRepo pf = ((PessoaFisicaRepo) repo).obter(id);
   if(pf!=null){
    System.out.println("Digite o CPF antigo: " + pf.getCpf());
    System.out.println("Digite o novo CPF do usuário: ");
    pf.setCpf(reader.readLine());
    System.out.println("Digite a Idade antiga: " + pf.getIdade());
    System.out.println("Digite a nova Idade do usuário: ");
    pf.setIdade(Integer.parseInt(reader.readLine()));
    System.out.println("Digite o Nome Antigo: " + pf.getNome());
    System.out.println("Digite o novo Nome do usuário: ");
    pf.setNome(reader.readLine());
    ((PessoaFisicaRepo)repo).alterar(pf);
    System.out.println("Usuário alterado com sucesso.");
    }else{
      System.out.println("Desculpe.Usuário não encontrado.Tente novamente!!!");}
   }
   else if (repo instanceof PessoaJuridicaRepo){
    PessoaFisicaRepo pj = ((PessoaFisicaRepo)repo).obter(id);
    if(pj != null){
    System.out.println("CNPJ Antigo: " + pj.getCnpj());
    System.out.println("Digite o novo CNPJ do usuário: ");
    pj.setCnpj(reader.readLine());
    System.out.println("Digite o nome da empresa antiga" + pj.getNomeEmpresa());
    System.out.println("Digite o novo nome da empresa: ");
    pj.setNomeEmpresa(reader.readLine());
    ((PessoaJuridicaRepo)repo).alterar(pj);
    System.out.println("Usuário Empresa, foi alterado com sucesso.");
    }
     else{
      System.out.println("Desculpe. Usuário ou empresa não encontrada. Tente novamente!");
    }
    }
}
```

main.java

model > gerenciador > main.java

```java
    private static void excluirPessoa(Object repo, BufferedReader reader) throws IOException {
      System.out.println(" Digite o Id do usuário que você deseja excluir:  ");
      int id = Integer.parseInt(reader.readLine());
      if (repo instanceof PessoaFisicaRepo) {
      ((PessoaFisicaRepo)repo).excluir(id);
      System.out.println("Usuário excluído com sucesso.");
      }
       else if(repo instanceof PessoaJuridicaRepo) {
       ((PessoaFisicaRepo)repo).excluir(id);
       System.out.println("Usuário Empresa foi excluído com sucesso.");
       }
    }

    private static void buscarPessoa(Object repo, BufferedReader reader) throws IOException {
      System.out.println(" Digite o Id para procurar o usuário: ");
      int id = Integer.parseInt(reader.readLine());
      if (repo instanceof PessoaFisicaRepo){
       PessoaFisicaRepo pf = ((PessoaFisicaRepo)repo).obter(id);
       if(pf != null){
      System.out.println("Id: " + pf.getId() );
      System.out.println("CPF: " + pf.getCpf());
      System.out.println("Idade: " + pf.getIdade());
      System.out.println("Nome: " + pf.getNome());
      }
       else{System.out.println("Desculpe! Usuário não encontrado. Tente novamente.");}
      }
    }
      else if (repo instanceof PessoaJuridicaRepo){
        int id;
        PessoaJuridicaRepo repo;
        PessoaJuridica pj = ((PessoaJuridicaRepo)repo).obter(id);
        System.out.println("Id: " + pj.getId());
        System.out.println("CNPJ: " + pj.getCnpj());
        System.out.println("Nome: " + pj.getNome());
      }
```

main.java

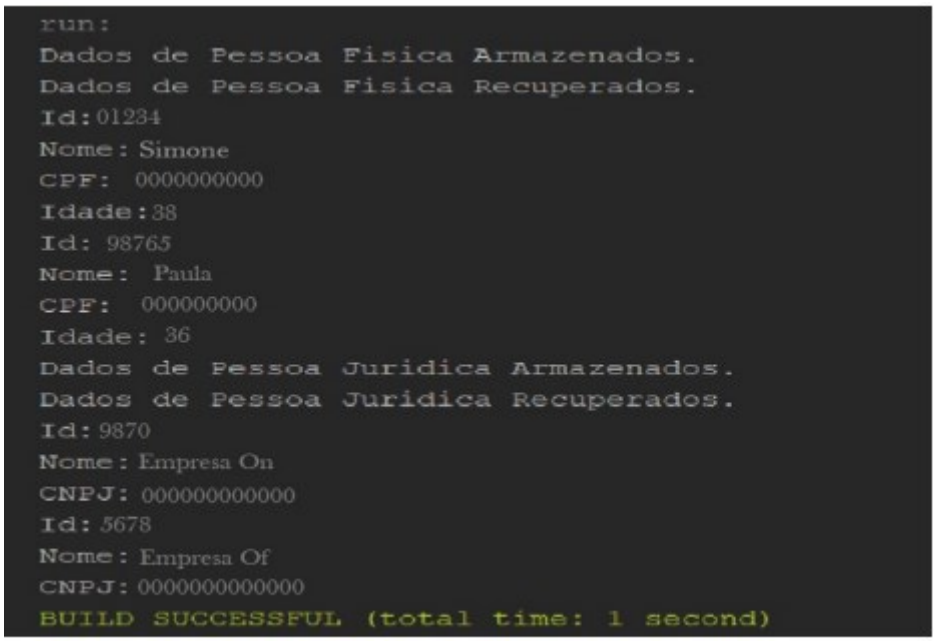
model > gerenciador > main.java

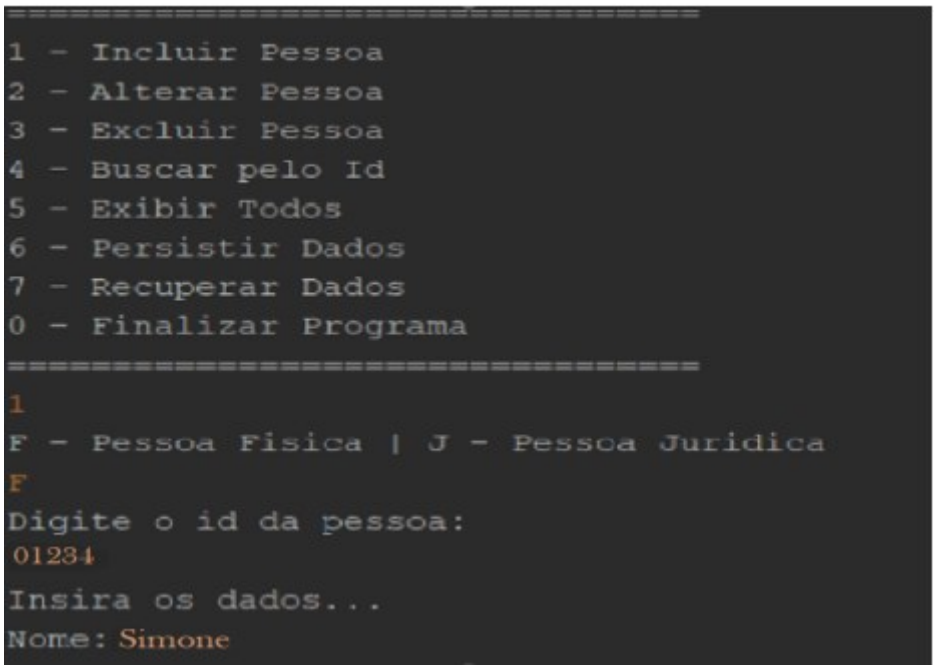
```java
      else {
      System.out.println("Desculpe. Usuário Empresa não encontrado. Tente novamente.");
      }
    private static void exibirTodasPessoas(Object repo){
      if(repo instanceof PessoaFisicaRepo){
      for(PessoaFisicaRepo pf :((PessoaFisicaRepo)repo). obterTodos()){
      System.out.println("id: " + pf.getId());
      System.out.println("Nome: " + pf.getNome());
      System.out.println("CPF: " + pf.getCpf());
      }
      }
       else if(repo instanceof PessoaJuridicaRepo){
       for(PessoaJuridicaRepo pj : ((PessoaJuridicaRepo)repo).obterTodos()){
       System.out.println("id: " + pj.getId());
       System.out.println("Nome: " + pj.getNome());
       System.out.println("CNPJ: " + pj.getCnpj());
       }
       }
    }

    private static void exibirTodasPessoas(PessoaJuridicaRepo repojuridica, BufferedReader reader)
    private static void exibirTodasPessoas(PessoaFisicaRepo repoFisica, BufferedReader reader) {}
    private static void pesquisarPessoa(PessoaJuridicaRepo repojuridica, BufferedReader reader) {}
    private static void pesquisarPessoa(PessoaFisicaRepo repoFisica, BufferedReader reader) {}
    private static void excluirPessoa(PessoaJuridicaRepo repojuridica, BufferedReader reader) {}
    private static void excluirPessoa(PessoaFisicaRepo repoFisica, BufferedReader reader) {}
    private static void alterarPessoa(PessoaJuridicaRepo repojuridica, BufferedReader reader) {}
    private static void alterarPessoa(PessoaFisicaRepo repoFisica, BufferedReader reader) {}
```


main.java
model > gerenciador > main.java
300 private static void excluirPessoa(PessoaFisicaRepo repoFisica, BufferedReader reader) {}
301 private static void alterarPessoa(PessoaJuridicaRepo repojuridica, BufferedReader reader) {}
302 private static void alterarPessoa(PessoaFisicaRepo repoFisica, BufferedReader reader) {}
303
304
305 public static BufferedReader getReader() {
306 return reader;}
307 public static void setReader(BufferedReader reader) {}
308 main.reader = reader;
309 public static String getNome() {
310 return nome;}
311 public static void setNome(String nome) {
312 main.nome = nome;}
313 public static Object getCnpj() {
314 return cnpj;}
315 public static void setCnpj(Object cnpj) {
316 main.cnpj = cnpj;}
317
318 @Override
319 public String toString() {
320 return "main []";
321 }
322 }
323

Resultado da execução dos códigos:

Incluir os resultados obtidos ao executar o programa, como a interação com o usuário, os dados persistidos e recuperados, e qualquer saída relevante do sistema.





Análise:

Os elementos estáticos em Java são relação à classe. O modificador Static é usado para declarar elementos estáticos, incluindo variáveis, métodos e blocos.

Variáveis estáticas fazem parte da classe e são compartilhadas por todas as instâncias dessa classe.

Métodos estáticos pertencem à classe e não a instâncias específicas.

Scanner é utilizada para facilitar a leitura de dados da entrada padrão, comumente associada ao teclado. Ela pertence ao pacote java.util, principal função é simplificar a interação do programa com o usuário, fornece métodos que permitem a leitura de diferentes tipos de dados, como Strings, inteiros e vários outras tipos interação com usuário.

As classes dos repositórios, tem a característica de organizar e modular, deixando o código mais intuitivo e fácil de dar manutenção e entendimento.

Conclusão:

A utilização dos elementos estáticos, classes de repositórios e classe Scanner, proporciona um código organizado, interativo, modular e com logica de persistência e uma ótima estrutura e desenvolvimento de sistemas mais complexos e eficientes em Java.