

# Relatório Discente

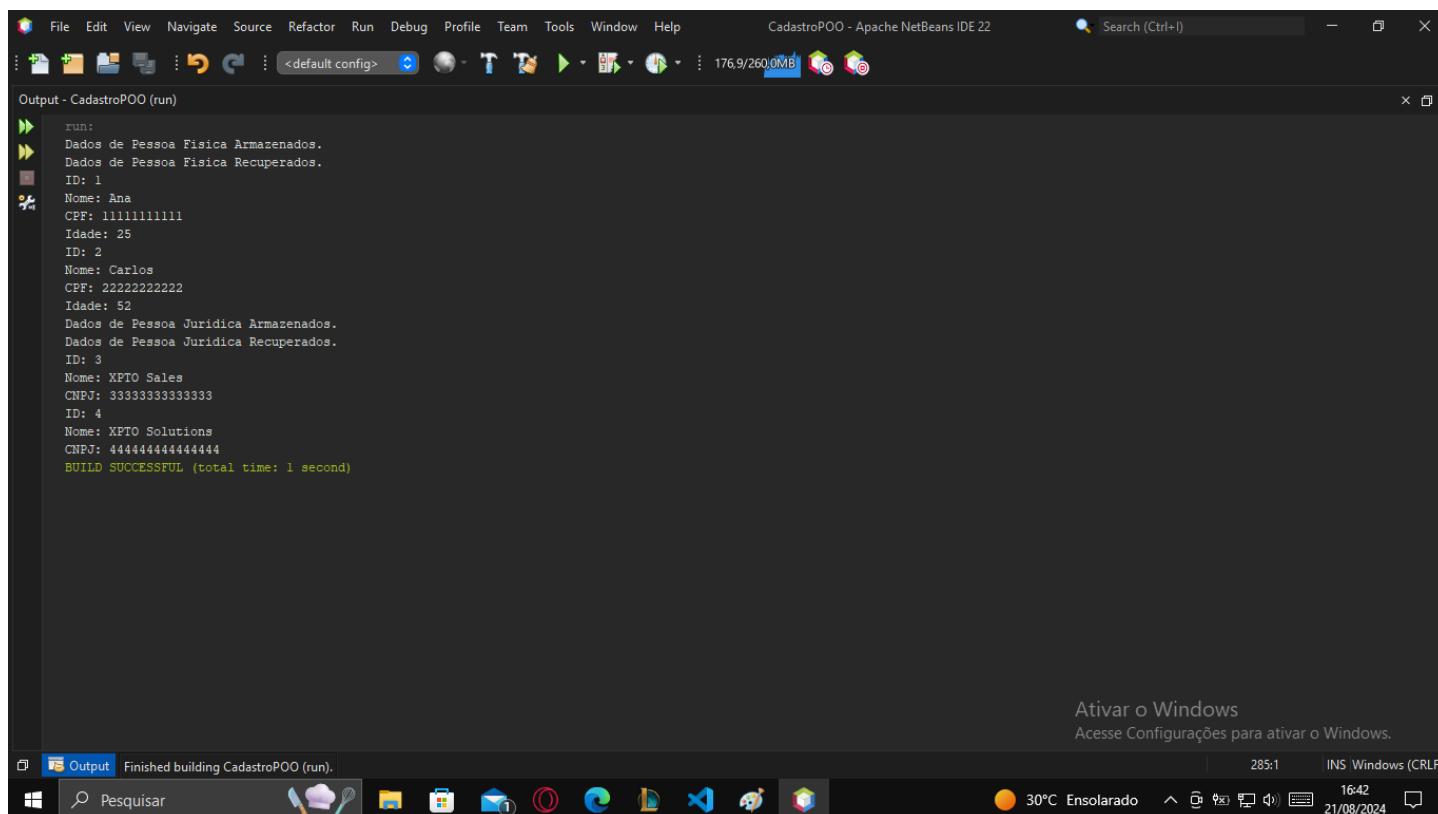
## Desenvolvimento Full stack

- Campus Virtual
- Nível 1: Iniciando o Caminho Pelo Java
- Período 2024.3
- Nome: Gabriel Gonçalves Silva Costa
- Link:

### RPG0014 - Iniciando o caminho pelo Java

Os códigos estão no Github através do link fornecido.

## Resultado 1:



```
run:
Dados de Pessoa Fisica Armazenados.
Dados de Pessoa Fisica Recuperados.
ID: 1
Nome: Ana
CPF: 111111111111
Idade: 25
ID: 2
Nome: Carlos
CPF: 222222222222
Idade: 52
Dados de Pessoa Juridica Armazenados.
Dados de Pessoa Juridica Recuperados.
ID: 3
Nome: XPTO Sales
CNPJ: 33333333333333
ID: 4
Nome: XPTO Solutions
CNPJ: 44444444444444
BUILD SUCCESSFUL (total time: 1 second)
```

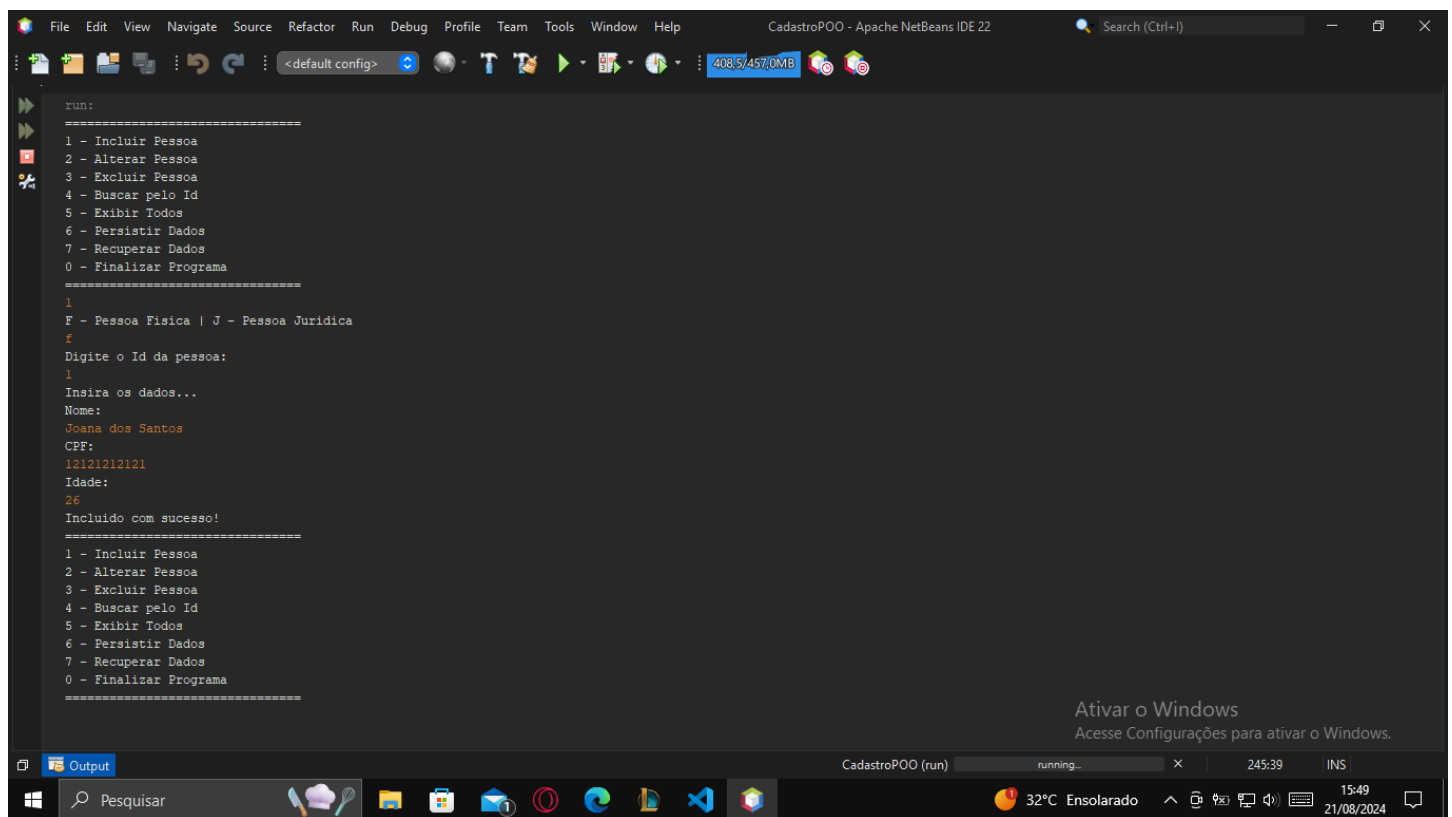
Usar herança facilita no uso das classes, na verdade poupa muito trabalho do programador e mantém o código limpo.

Eu só percebi de fato a importância de Serializable na hora de persistir e recuperar os arquivos. Já que não conseguia recuperá-los sem.

A API Stream permite processar coleções de dados de maneira declarativa, usando operações funcionais que são mais concisas e expressivas em comparação com as abordagens imperativas tradicionais.

"Data Access Object" (DAO)

## Resultado 2:



```
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
=====
1
F - Pessoa Fisica | J - Pessoa Juridica
f
Digite o Id da pessoa:
1
Insira os dados...
Nome:
Joana dos Santos
CPF:
12121212121
Idade:
26
Incluido 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
=====
```

**elementos estáticos** são componentes que pertencem à classe em vez de instâncias individuais da classe. Por várias razões main é declarado como método estático, as três são: Ponto de entrada do programa, Acesso sem instância e ele é simples e direto fazendo assim ele ser executado de maneira eficiente e previsível.

**Scanner** permite que eu receba informações do usuário por meio de inputs.

As classes repositório foram de ajuda na organização do código, também o deixou mais funcional e bem limpo. É uma prática pra se manter em mente sempre.