

# TRABALHO 1

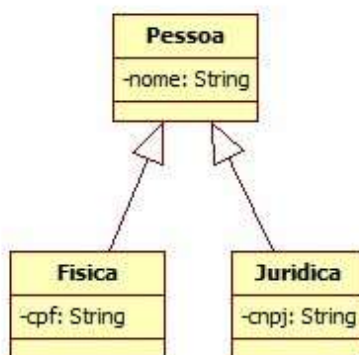
GRUPO: 3 PESSOAS / PONTO: 1,5

## HERANÇA/POLIMORFISMO/COLEÇÕES

### AVISO IMPORTANTE

- OS TRABALHOS QUE TIVEREM INDICÍDIOS DE CÓPIA TERÃO SUA NOTA ZERADA, ESTA REGRA VALE PARA O TRABALHO ORIGINAL E AS RESPECTIVAS CÓPIAS. APROVEITEM O DESAFIO PARA APRENDER OOP.
- APENAS UMA PESSOA DO GRUPO DEVERÁ POSTAR A ATIVIDADE.
- COLOCAR O NOME DOS INTEGRANTES DENTRO DO CÓDIGO FONTE DA APLICAÇÃO DE PREFERÊNCIA NO MÉTODO MAIN

Criar uma aplicação em JAVA que permita um determinado usuário realizar a leitura de dados de pessoas (físicas ou jurídicas), os dados das pessoas deverão ser incluídos em uma lista para impressão futura após a finalização da inclusão. Mais abaixo pode ser visto a estrutura de classes a ser utilizada e os dados de exemplos que serão necessários serem imputados para testar a aplicação.



Tipo	Nome	Documento
Física	MARIA DA SILVA	94636768051
Física	JOSÉ DA SILVA	16564904045
Jurídica	ACME DISTRIBUÍDORA	98379538000153
Jurídica	XPTO DISTRIBUÍDORA	43793891000193

Observação, os números contidos na coluna "Documento" são fictícios e foram gerados através dos links: [https://www.4devs.com.br/gerador\\_de\\_cnpj](https://www.4devs.com.br/gerador_de_cnpj) e [https://www.4devs.com.br/gerador\\_de\\_cpf](https://www.4devs.com.br/gerador_de_cpf), caso desejarem pode utilizar outras informações.

#### Detalhes técnicos de implementação:

1. Utilizar conceito de classes;
2. Utilizar conceito de construtores;
3. Utilizar conceito de métodos;
4. Utilizar conceito de herança;
5. Utilizar conceito de polimorfismo;
6. Utilizar conceito de listas;
7. Realizar a leitura de dados via aplicação console, não há limite para digitação de dados, o usuário pode incluir quantas pessoas ele quiser. Dica: utilizar uma estrutura de repetição para que o usuário informe quando ele deseja parar de cadastrar pessoas;
8. Dados de pessoas físicas e jurídicas deverão ser armazenados em uma ÚNICA lista;
9. A classe de pessoa física deverá ter a possibilidade de realizar a validação do CPF para verificar se o número informado é válido (pesquisar exemplos de algoritmos em JAVA para validação de CPF);
10. A classe de pessoa jurídica deverá ter a possibilidade de realizar a validação do CNPJ para verificar se o número informado é válido (pesquisar exemplos de algoritmos em JAVA para validação de CNPJ);
11. Após o usuário informar que não deseja mais cadastrar pessoas o aplicativo deverá realizar a impressão dos dados armazenados na lista, este deverá percorrer por todos os itens e imprimir a informação contida nas classes seguindo o exemplo para PESSOA FÍSICA: "**Nome: <nome> - CPF: <documento> - STATUS: <status>**" e o seguinte exemplo para PESSOA JURÍDICA: "**Nome: <nome> - CNPJ: <documento> - STATUS: <status>**".
  - a. Observação:
    - i. **<nome>**: deverá apresentar o nome da pessoa cadastrada;
    - ii. **<documento>**: deverá apresentar o número do CPF (pessoa física) ou CNPJ (pessoa jurídica) cadastrada;
    - iii. **<status>**: deverá apresentar as informações de VÁLIDO ou INVÁLIDO, para pessoas físicas validar se o CPF é válido, para pessoas jurídicas validar se o CNPJ é válido.
  - b. Exemplo de mensagem impressa para pessoas físicas:  
**Nome: JOSE DA SILVA - CPF: 12345678901 - STATUS: VÁLIDO**
  - c. Exemplo de mensagem impressa para pessoas jurídicas:  
**Nome: ACME GAMES - CNPJ: 11222333444455 - STATUS: INVÁLIDO**