



# Universidade Estácio de Juazeiro da Bahia

## Desenvolvimento Full Stack

### Iniciando o caminho pelo Java

#### Relatório discente de acompanhamento

**Nome:** Gerson José de Almeida Júnior

**Número da Turma:** 23.3

**Semestre Letivo:** 3º Semestre

**Github:** <https://github.com/gersonjose9713/Missao-Pratica-Nivel-1-Mundo-3>

## Criação das Entidades e Sistema de Persistência

#### Objetivo da Prática:

#### Os objetivos da prática incluem:

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.
5. Implementar um sistema cadastral em Java, utilizando os recursos da programação orientada a objetos e a persistência em arquivos binários.

#### Resultados da Execução dos Códigos:

```
Run Main x
Nome: Maria
Id: 2
CPF: 25504595983
Idade: 30
Nome: Caroline
Id: 3
CPF: 77175538804
Idade: 40
Nome: Pedro
Id: 1
CPF: 90589504509
Idade: 60
```

```
Run Main x
Nome: T&F Telecomunicações Ltda
Id: 1
CNPJ: 80.186.933/0001-54
Nome: Edmundo Locações de Automóveis
Id: 2
CNPJ: 64.759.835/0001-02
Nome: Lorenzo Restaurante Ltda
Id: 3
CNPJ: 17.765.487/0001-04
```

## **Análise e Conclusão:**

### **Quais as vantagens e desvantagens do uso de herança?**

O uso de herança na programação orientada a objetos oferece a vantagem de promover a reutilização de código, permitindo que classes derivadas herdem propriedades e comportamentos de classes base. No entanto, pode levar a um acoplamento excessivo entre classes e tornar o código mais complexo. Além disso, a herança é uma relação de "é um", o que significa que deve representar uma relação verdadeira entre as classes.

### **Por que a interface Serializable é necessária ao efetuar persistência em arquivos binários?**

A interface Serializable é necessária ao efetuar a persistência em arquivos binários porque ela marca uma classe como "serializável", o que significa que seus objetos podem ser convertidos em bytes e posteriormente reconstruídos. Isso é essencial ao salvar e recuperar objetos Java em arquivos binários, como no caso da persistência de dados em disco.

### **Como o paradigma funcional é utilizado pela API stream no Java?**

A API Stream do Java utiliza o paradigma funcional para operações em coleções de dados. Ela permite a aplicação de funções lambda e operações de alto nível, como mapeamento, filtragem e redução, de forma concisa e eficiente. Isso torna o código mais legível e facilita o processamento de grandes conjuntos de dados.

### **Quando trabalhamos com Java, qual padrão de desenvolvimento é adotado na persistência de dados em arquivos?**

Ao trabalhar com Java, um padrão comum de desenvolvimento na persistência de dados em arquivos é o uso de objetos serializados para armazenar e recuperar informações. Isso permite que os objetos sejam facilmente gravados em arquivos binários e posteriormente lidos e reconstruídos. No entanto, é importante considerar a compatibilidade entre versões ao usar a serialização e tratar exceções durante a leitura e gravação de arquivos.