

**UNIVERSIDADE ESTÁCIO DE SÁ**

**CAMPUS POLO CENTRO - MARICÁ - RJ**

**ALUNO:** LUIZ CARLOS MARINHO JUNIOR

**MATRÍCULA:** 2023.11.17557-1

**CURSO:** DESENVOLVIMENTO FULL STACK

**SEMESTRE LETIVO:** 2024.4

**DISCIPLINA:** NÍVEL 1: INICIANDO O CAMINHO PELO JAVA

**TÍTULO: 2º PROCEDIMENTO | CRIAÇÃO DO CADASTRO EM MODO TEXTO**

RIO DE JANEIRO

2024

LUIZ CARLOS MARINHO JUNIOR

**1º PROCEDIMENTO | CRIAÇÃO DAS ENTIDADES E SISTEMA DE PERSISTÊNCIA**

Trabalho prático para aprovação na disciplina de Nível 1: Iniciando o Caminho Pelo Java.

Tutora: Prof. Maria B.

RIO DE JANEIRO

2024

**SUMÁRIO**

[**1. OBJETIVOS DA PRÁTICA** 4](#_Toc9)

[**2. CÓDIGOS SOLICITADOS** 4](#_Toc10)

[**2.1. APRESENTAÇÃO** 4](#_Toc11)

[**2.2. PRINCIPAIS INSTRUÇÕES DO 2ª PROCEDIMENTO** 4](#_Toc12)

[**3. RESULTADO DA EXECUÇÃO DOS CÓDIGOS** 4](#_Toc13)

[**4. ANÁLISE E CONCLUSÃO** 5](#_Toc14)

[**4.1. O QUE SÃO ELEMENTOS ESTÁTICOS E QUAL O MOTIVO PARA O MÉTODO MAIN ADOTAR ESSE MODIFICADOR?** 5](#_Toc15)

[**4.2. PARA QUE SERVE A CLASSE SCANNER?** 5](#_Toc16)

[**4.3. COMO O USO DE CLASSES DE REPOSITÓRIO IMPACTOU NA ORGANIZAÇÃO DO CÓDIGO?** 5](#_Toc17)

# **1. 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;
5. No final do projeto, terá implementado um sistema cadastral em Java, utilizando os recursos da programação orientada a objetos e a persistência em arquivos binários.

**2. CÓDIGOS SOLICITADOS**

## **2.1. APRESENTAÇÃO**

Todos os códigos que serão aqui apresentados, estão disponíveis de forma completa no repositório do Github. O desenvolvimento do código está separado por branch, e no caso desse 1ª Procedimento, o código se encontra na branch PrimeiroProcedimento.

O **link para o repositório do Github** desse 1ª Procedimento se encontra logo abaixo:  
<https://github.com/luizmarinhojr/cadastro-poo/tree/PrimeiroProcedimento>.

## **2.2. PRINCIPAIS INSTRUÇÕES DO 2ª PROCEDIMENTO**

# **3. RESULTADO DA EXECUÇÃO DOS CÓDIGOS**

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

# **4. ANÁLISE E CONCLUSÃO**

## **4.1. O QUE SÃO ELEMENTOS ESTÁTICOS E QUAL O MOTIVO PARA O MÉTODO MAIN ADOTAR ESSE MODIFICADOR?**

## **4.2. PARA QUE SERVE A CLASSE SCANNER?**

## **4.3. COMO O USO DE CLASSES DE REPOSITÓRIO IMPACTOU NA ORGANIZAÇÃO DO CÓDIGO?**

**REFERÊNCIAS**

ORACLE, Java™ Platform, Standard Edition 8 API Specification. Disponível em: [<https://docs.oracle.com/javase/8/docs/api/](https://docs.oracle.com/javase/8/docs/api/)>. Acesso em 09 de dezembro de 2024.

BAELDUNG, How to Serialize a Singleton in Java. Disponível em: [<https://www.baeldung.com/java-serialize-singleton>.](https://www.baeldung.com/java-serialize-singleton.) Acesso em 09 de dezembro de 2024.

CODEHS, Data Persistence in Java. Disponível em: <<https://codehs.com/tutorial/david/data-persistence-in-java>>. Acesso em 09 de dezembro de 2024.

MEDIUM, Programação Funcional no Java. Disponível em: [<https://medium.com/@nvieirarafael/programa%C3%A7%C3%A3o-funcional-no-java-2a005964cb20](https://medium.com/@nvieirarafael/programa%C3%A7%C3%A3o-funcional-no-java-2a005964cb20)>. Acesso em 09 de dezembro de 2024.