



# Linguagem Orientada a Objetos

Encontro 03 – Pilares, Classes e Objetos

Prof. Luiz Augusto Rodrigues  
[luiz.a.rodrigues@cogna.com.br](mailto:luiz.a.rodrigues@cogna.com.br)

# Avisos Importantes



**Nenhum** material será enviado via e-mail. Os materiais serão disponibilizados no **AVA** e no google drive:  
<https://bit.ly/47qzfvI>



Dúvidas, questionamentos, entre outros deverão ser realizados pelo **e-mail** e pelo **Whatsapp** da disciplina.



Para ingressar no grupo do **Whatsapp** da disciplina acesse o link a seguir e selecione sua disciplina.  
[linklist.bio/profluizao\\_2023-2](https://linklist.bio/profluizao_2023-2)



A Disciplina de Programação Orientada a Objetos, a partir deste slide, será referenciada pela sigla **LOO**.



# Assuntos Abordados

Neste encontro, debateremos os seguintes tópicos:

- Revisão – Pilares
- Aula Prática
  - Diagrama de Classe
  - Pilares
  - Classes
  - Objetos
  - Atributos
  - Propriedades
  - Métodos Construtores

# LOO

## Encontro 03

---

**Instruções Importantíssimas!!!**

# Instruções Importantíssimas!!!

Para este semestre, na disciplina de POO, utilizaremos OBRIGATORIAMENTE as seguintes ferramentas:

- Sistema Operacional Windows 10, Linux ou MacOS.
- Linguagem de Programação Java.
- JDK 17 ou superior
  - [https://download.oracle.com/java/17/archive/jdk-17.0.6\\_windows-x64\\_bin.exe](https://download.oracle.com/java/17/archive/jdk-17.0.6_windows-x64_bin.exe)
- VS Code.

# Instruções Importantíssimas!!!

- Git 2.39.2 ou superior
  - <https://github.com/git-for-windows/git/releases/download/v2.39.2.windows.1/Git-2.39.2-64-bit.exe>
- Github
  - <https://github.com/>
  - <https://desktop.github.com/>
- Tutorial em Git e Github
  - Tutorial Git e Github 2022 – Introdução prática para iniciantes
  - Canal DevSuperior
  - <https://youtu.be/hZf1teRFNg>

Quem for utilizar Linux ou MacOSx, fale comigo após a aula, ou pelo grupo do WhatsApp, para instruções de instalação.

# Instruções Importantíssimas!!!

- Os relatórios não serão utilizados nessa disciplina, por se tratar de uma disciplina 90% prática.
- No lugar, teremos atividades de pós-aula, que deverão ser respondidas e justificadas, para fixação do conteúdo discutido na aula.
- As atividades de aula estarão disponíveis apenas no Github da turma.
- O tutorial do Git e Github é **obrigatório**.

# LOO

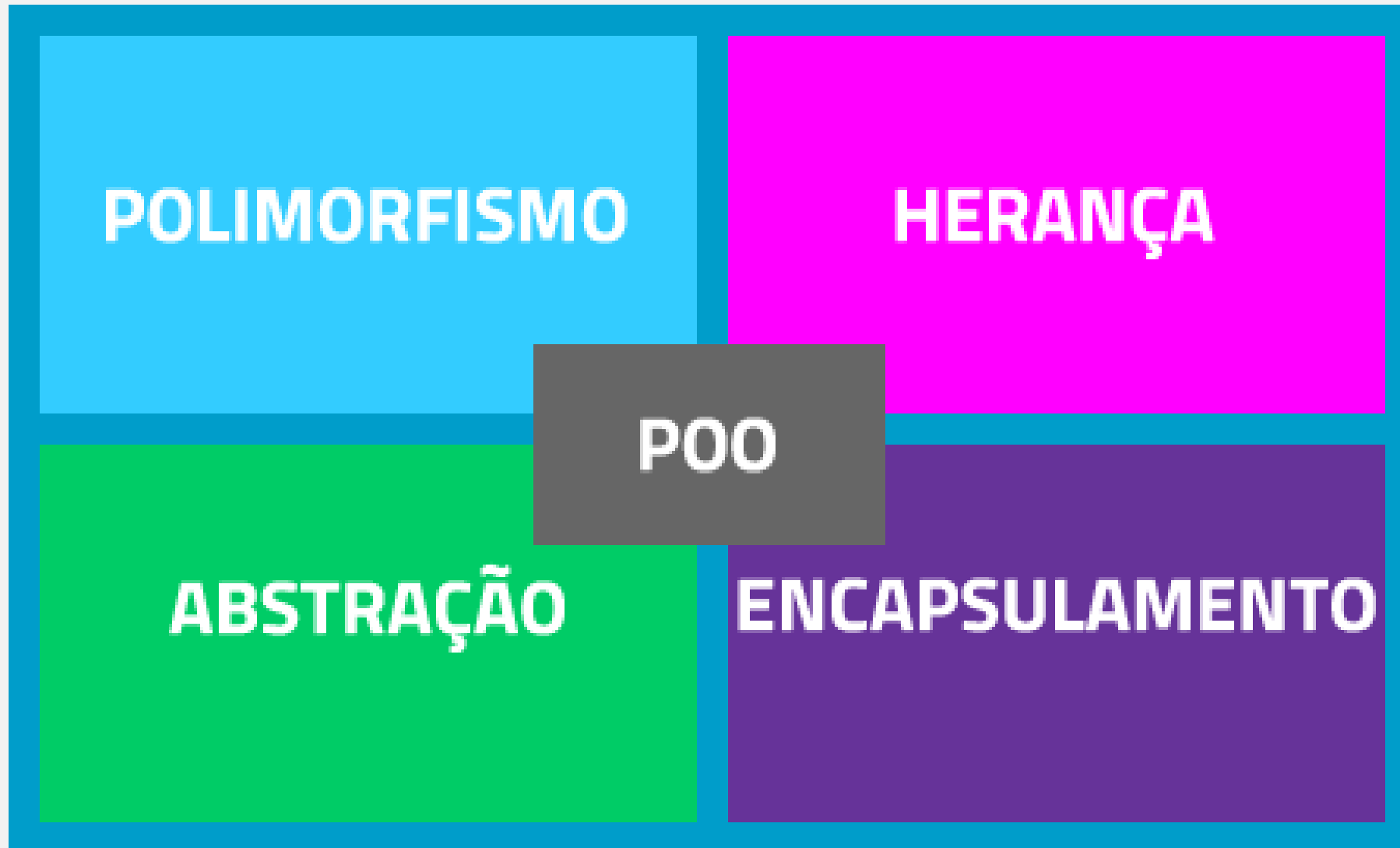
## Encontro 03

---

Revisão – Pilares



# Princípios de POO



# LOO

## Encontro 03

---

Diagrama de Classe

# Diagrama de Classe

Em UML, diagramas de classes são um dos seis tipos de diagramas estruturais. Os diagramas de classe são fundamentais para o processo de modelagem de objetos e modelam a estrutura estática de um sistema.

Dependendo da complexidade de um sistema, é possível utilizar um único diagrama de classe para modelar um sistema inteiro ou vários diagramas de classe para modelar os componentes de um sistema.

# Diagrama de Classe

Os diagramas de classe são as cópias do sistema ou subsistema.

Você pode utilizar os diagramas de classe para modelar os objetos que compõem o sistema, para exibir os relacionamentos entre os objetos e para descrever o que esses objetos fazem e os serviços que eles fornecem.

Os diagramas de classe são úteis em muitos estágios do design do sistema.

No estágio de análise, um diagrama de classe pode ajudá-lo a compreender os requisitos do domínio do problema e a identificar seus componentes.

# Diagrama de Classe

Em um projeto de software orientado a objetos, os diagramas de classe criados durante os estágios iniciais do projeto contêm classes que normalmente são convertidas em classes e objetos de software reais quando você grava o código.

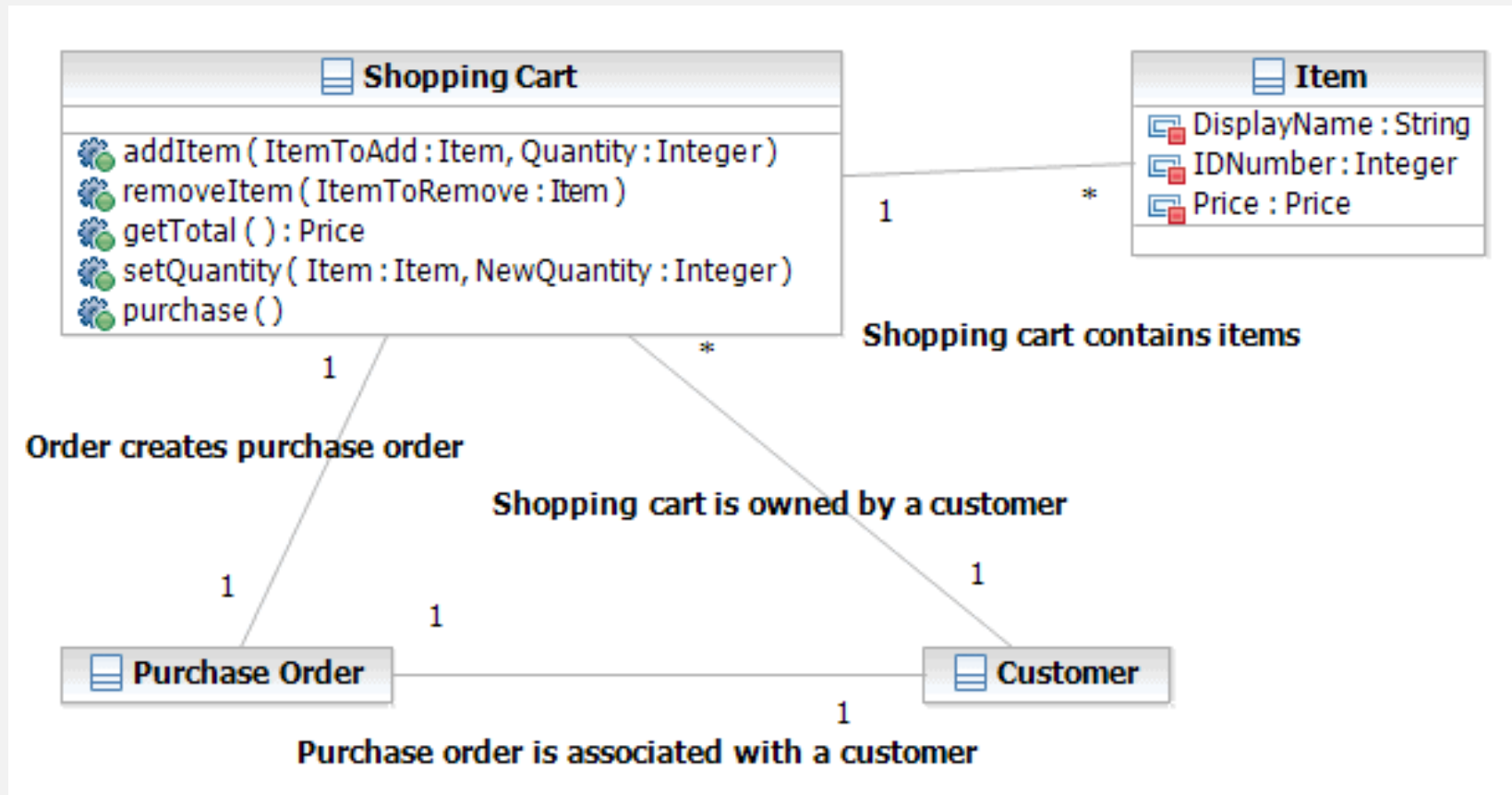
Posteriormente, é possível refinar a análise e os modelos conceituais anteriores em diagramas de classe que mostrem as partes específicas do sistema, interfaces com o usuário, implementações lógicas e assim por diante.



# Diagrama de Classe

Os diagramas de classe tornam-se, então, uma captura instantânea que descreve exatamente como o sistema funciona, os relacionamentos entre os componentes do sistema em vários níveis e como planeja implementar esses componentes.

# Diagrama de Classe



# Diagrama de Classe

Os seguintes tópicos descrevem elementos de modelos nos diagramas de classes:

## Classes

Em UML, uma classe representa um objeto ou um conjunto de objetos que compartilham uma estrutura e um comportamento comuns. As classes, ou instâncias de classes, são elementos de modelo comuns nos diagramas UML.

# Diagrama de Classe

## Objetos

Nos modelos UML, os objetos são elementos de modelo que representam instâncias de uma classe ou de classes. Você pode incluir objetos no modelo para representar instâncias concretas e prototípicas. Uma instância concreta representa uma pessoa ou coisa real no mundo verdadeiro. Por exemplo, uma instância concreta de uma classe **Customer** representa um cliente real. Uma instância prototípica de uma classe **Customer** contém dados que representam um cliente típico.

# Diagrama de Classe

## Sinais

Nos modelos UML, sinais são elementos do modelo independentes dos classificadores que os manipulam. Os sinais especificam comunicações assíncronas de uma via entre objetos ativos.

## Enumerações

Em modelos UML, enumerações são elementos do modelo em diagramas de classes que representam tipos de dados definidos pelo usuário. As enumerações contêm conjuntos de identificadores nomeados que representam os valores da enumeração. Esses valores são chamados de literais de enumeração.



# Diagrama de Classe

## Tipos de Dados

Em diagramas UML, tipos de dados são elementos de modelos que definem valores de dados. Você geralmente usa tipos de dados para representar tipos primitivos, como tipos inteiros ou de cadeia, e enumerações, como tipos de dados definidos pelo usuário.

## Artefatos

Em modelos UML, artefatos são elementos de modelo que representam as entidades físicas em um sistema de software. Os artefatos representam unidades físicas de execução, como por exemplo arquivos executáveis, bibliotecas, componentes de software, documentos e bancos de dados.

# Diagrama de Classe

## Relacionamentos em Diagramas de Classe

Na UML, um relacionamento é uma conexão entre elementos de modelo. Um relacionamento UML é um tipo de elementos de modelo que inclui semântica em um modelo, definindo a estrutura e o comportamento entre os elementos de modelo.

# Diagrama de Classe

## Qualificadores em Extremidades da Associação

Na UML, qualificadores são propriedades de associações binárias e são uma parte opcional de extremidades de associação. Um qualificador mantém uma lista de atributos de associações, cada um com um nome e um tipo. Os atributos de associação modelam chaves que são usadas como um subconjunto de instâncias de relacionamentos.

# LOO

## Encontro 03

---

Aula Prática

# Aula Prática

Vamos criar nosso programa Java.

Etapas da aula de hoje:

- Usando o VS Code
- Criando um projeto Java
- Criando o pacote de Domínio
- Criando as classes do Exemplo 01.
- Criando as classes do Exemplo 02.
- Otimizando as classes como no Exemplo 03.



# Aula Prática

1. Criar diagrama de classe Exemplo 01.
  1. Classe
  2. Atributos
  3. Propriedades
  4. Métodos Construtores

Pessoa
<ul style="list-style-type: none"><li>- codigo: int</li><li>- nome: string</li><li>- endereco: string</li><li>- telefone: string</li><li>- dataNascimento: data</li><li>- rg: string</li><li>- cpf: string</li><li>- dataInsercao: data</li></ul>
<ul style="list-style-type: none"><li>+ Pessoa()</li><li>+ Pessoa(int, string, string, string, data, string, string, data)</li></ul>

# Aula Prática

## 2. Criar diagrama de classe Exemplo 02.

1. Novas Classes
2. Atributos
3. Propriedades
4. Métodos Construtores
5. Análise e Verificação para possíveis

Pessoa
<ul style="list-style-type: none"><li>- codigo: int</li><li>- nome: string</li><li>- endereco: string</li><li>- telefone: string</li><li>- dataNascimento: data</li><li>- rg: string</li><li>- cpf: string</li><li>- dataInsercao: data</li></ul>
<ul style="list-style-type: none"><li>+ Pessoa()</li><li>+ Pessoa(int, string, string, string, data, string, string, data)</li></ul>

Cliente
<ul style="list-style-type: none"><li>- codigo: int</li><li>- nome: string</li><li>- endereco: string</li><li>- telefone: string</li><li>- dataNascimento: data</li><li>- rg: string</li><li>- cpf: string</li><li>- dataInsercao: data</li></ul>
<ul style="list-style-type: none"><li>+ Cliente()</li><li>+ Cliente(int, string, string, string, data, string, string, data)</li></ul>

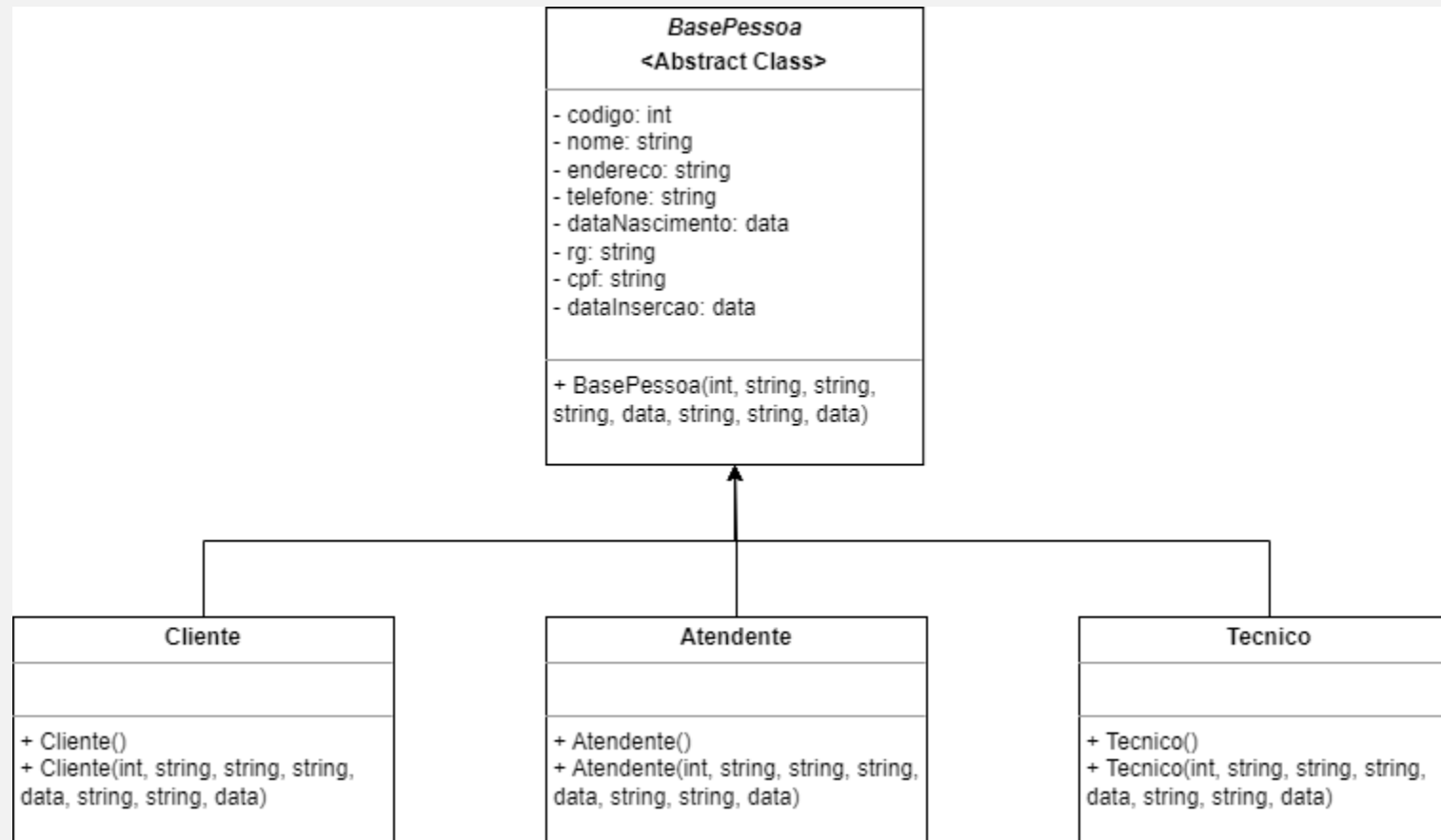
Atendente
<ul style="list-style-type: none"><li>- codigo: int</li><li>- matricula: int</li><li>- nome: string</li><li>- endereco: string</li><li>- telefone: string</li><li>- dataNascimento: data</li><li>- rg: string</li><li>- cpf: string</li><li>- dataInsercao: data</li></ul>
<ul style="list-style-type: none"><li>+ Cliente()</li><li>+ Cliente(int, string, string, string, data, string, string, data)</li></ul>

Tecnico
<ul style="list-style-type: none"><li>- codigo: int</li><li>- matricula: int</li><li>- nome: string</li><li>- endereco: string</li><li>- telefone: string</li><li>- dataNascimento: data</li><li>- rg: string</li><li>- cpf: string</li><li>- dataInsercao: data</li></ul>
<ul style="list-style-type: none"><li>+ Cliente()</li><li>+ Cliente(int, string, string, string, data, string, string, data)</li></ul>

# Aula Prática

## 3. Criar diagrama de classe Exemplo 03.

### 1. Otimizar código aplicando Pilares POO



# LOO

## Encontro 03

---

Atividade de Aula

# Atividade de Aula

1. Responder a atividade, denominada “LOO – Atividade 03.pdf”, disponível apenas no Github e no Google Drive. VALE NOTA!!!

2. Enviar as respostas justificadas no e-mail do professor, em formato PDF, com o seguinte assunto:

**[LOO] ATIVIDADE 03**

3. Data de entrega: 28/08/2023, até às 23h00.

# Encerramento



Dúvidas e sugestões, entre em contato pelo whatsapp da disciplina, ou mande um e-mail para [luiz.a.rodriques@cogna.com.br](mailto:luiz.a.rodriques@cogna.com.br)