

# POO2

---

## Encontro 01 - Revisão

**Professor Luiz Augusto Rodrigues**

***luiz.a.rodrigues@cogna.com.br***



# Sumário

- Pilares de POO
- Conceitos:
  - Classe
  - Objeto
  - Construtor
  - Método
  - Atributo
  - Propriedade
  - Variáveis
  - Instância de Objeto
  - Tipos de Variáveis
  - Classes Abstratas
  - Interfaces



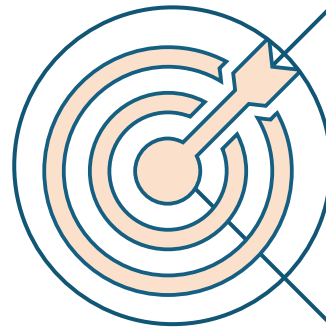
# Informações Importantes



**Nenhum** material será enviado via e-mail. Os materiais serão disponibilizados no **AVA** e no Google Drive da Disciplina.



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



A Disciplina de Programação Orientada a Objetos, a partir deste slide, será referenciada pela sigla **POO2**.

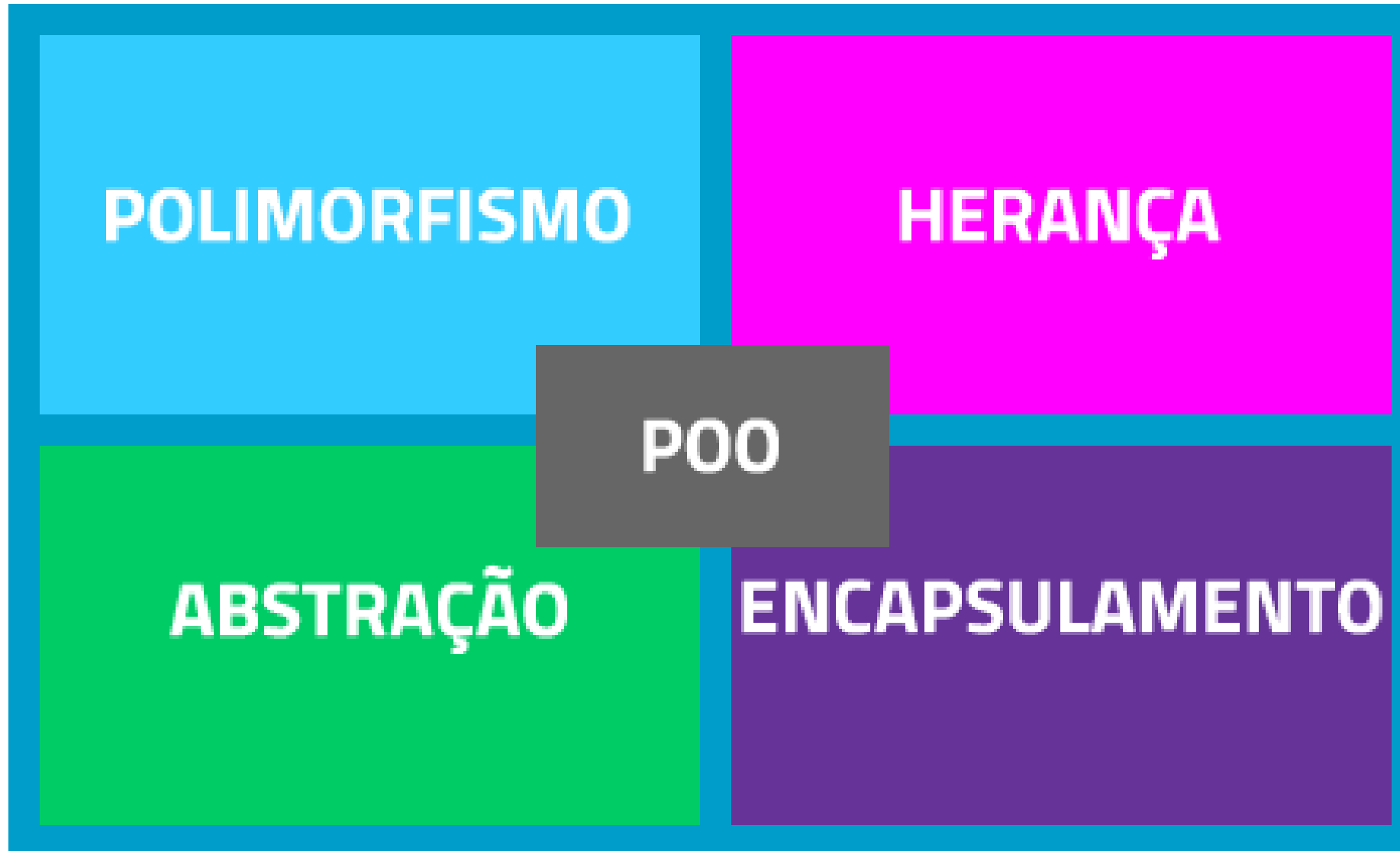
# Informações Importantes

**Grupo de Whatsapp da Disciplina:**

<https://tinyurl.com/mw2xdkd7>



# Pilares de POO



## ***Classes***

Classes são estruturas das linguagens de POO criadas para conter os dados que devem ser representados e as operações que devem ser efetuadas com esses dados para determinado modelo.

De modo mais direto, uma classe é uma estrutura que abstrai um conjunto de objetos com características similares, define o comportamento de seus objetos (através de métodos), e os estados possíveis destes objetos (através de atributos).

## *Objetos*

Objetos são instâncias de classes, que determinam qual informação um objeto contém e como ele pode manipulá-la.

É uma entidade capaz de reter um estado (informação) e que oferece uma série de informações (comportamento) ou para examinar ou para afetar este estado.

É através deles que praticamente todo o processamento ocorre em sistemas implementados com linguagens de programação orientada a objetos.

## *Atributos*

Atributos são características de um objeto, nos quais são armazenados os dados e informações relevantes ao objeto em si, seus tipos de dados.

Atributos podem ser privados (somente o próprio objeto conhece e manipula), públicos (acessíveis por qualquer objeto terceiro) ou protegido (apenas as classes derivadas, ou filhas, pode acessar e manipular).



## *Propriedades*

Propriedades são estruturas em uma classe que definem o acesso a informações dentro dela.

Uma propriedade é composta por dois métodos (os famosos Get e Set): um método para retornar a informação chamado Get, e um método para definir o valor da propriedade chamado Set.

Geralmente, para cada método existe uma variável dentro da classe que armazena o valor da propriedade.

## *Métodos*

Os métodos são procedimentos ou funções que realizam as ações próprias do objeto.

Assim, os métodos são as ações que o objeto pode realizar.

Tudo o que o objeto faz é através de seus métodos, pois é através deles que um objeto se manifesta, através deles que o objeto interage com os outros objetos.

## *Métodos Construtores*

Os construtores são tipos especiais de métodos usados para criar e inicializar objetos. É através deste tipo especial de método, chamado de construtor, que você cria instâncias de uma classe.

Assim, os construtores permitem que o programador crie objetos da classe e defina valores padrão, limite a instanciação e grave códigos flexíveis e fáceis de ler.

## *Instância de Objetos*

Uma instância de objeto nada mais é do que a criação de um objeto a partir de uma classe definida, através da chamada de seu método construtor.

Imagine que temos o objeto da classe Pessoa, e este objeto apenas representa uma pessoa. Se quisermos dar uma identidade a este objeto, temos que instanciá-lo, isto é, trazer esta pessoa à vida.

Criar um novo objeto também pode significar criar um novo carro, um novo botão, uma nova planilha.

## ***Classes Abstratas***

Uma classe abstrata é uma classe que serve de modelo para outras classes. Ela sempre será uma superclasse genérica, e suas subclasses serão mais específicas.

Além disso, ela não pode ser instanciada e pode conter ou não métodos abstratos, podendo ser implementados nas classes descendentes.

Ou seja, uma classe abstrata pode implementar ou não um método, sendo obrigatória a existência de pelo menos um método abstrato, sem corpo.

Ao ouvirmos a palavra animal, podemos imaginar um gato, um cachorro, etc. Ou seja, uma palavra genérica pode nos levar à imaginar coisas mais específicas.

## ***Classes Abstratas - Características***

- Uma classe abstrata é uma classe que não pode ser instanciada. Você não pode criar um objeto a partir de uma classe abstrata.
- Uma classe abstrata pode ser herdada e geralmente serve como classe base para outras classes.
- Uma classe abstrata pode conter métodos abstratos e métodos comuns. Também podem possuir construtores, propriedades, indexadores e eventos.
- Uma classe abstrata não pode ser estática (`static`).
- Uma classe abstrata não pode ser selada (`sealed`)(*final*).
- Uma classe abstrata pode herdar de outra classe abstrata.

## *Métodos Abstratos*

Um método indicado pela palavra ***abstract*** indica que o mesmo não possui código implementado, e deve ter sua implementação definida na próxima classe real da hierarquia.

- Um método abstrato é um método que não possui implementação na classe abstrata.
- Possui somente a definição de sua assinatura. A sua implementação deve ser feita na classe derivada.
- É por natureza um método virtual e deve ser implementado usando o modificador ***override***.
- Somente pode existir em uma classe abstrata.
- Não pode usar os modificadores ***static*** e ***virtual***.

## *Propriedades Abstratas*

Uma declaração de propriedade ***abstract*** não fornece uma implementação de acesso à propriedade – ela declara que a classe dá suporte às propriedades, mas deixa a implementação do acessor para classes derivadas.

Ou seja, nós simplesmente declaramos sua assinatura. O código de implementação deve obrigatoriamente ser desenvolvido na classe real mais próxima na hierarquia de herança.



## *Interfaces*

A interface é bastante semelhante com o conceito de Classes Abstratas, todavia, sua principal diferença é o fato de todos os seus métodos serem abstratos, isto é, temos somente assinaturas, nenhuma implementação.

Na classe abstrata poderia haver métodos não abstratos, ou seja, métodos definidos, com seus devidos códigos implementados, enquanto que em uma interface isso não é possível.

Além disso, uma interface não pode ser instanciada.

## *Interfaces - Características*

- A interface de um objeto consiste de um conjunto de métodos que um objeto deve suportar.
- Em interfaces todos os métodos ou propriedades são **públicos**.
- Uma interface pode ser compreendida como um contrato, contendo pré e pós condições, aqui compreendidos como métodos ou propriedades, as quais devem ser cumpridas pela classe que implementa esta interface, obrigatoriamente.
- Apesar de ter implementação obrigatória, se a classe em questão for uma classe abstrata, a implementação do contrato pode ser delegada às classes filhas.

## *Interfaces - Características*

- Ainda sobre classes abstratas, um método oriundo de interface pode ser implementado e declarado como virtual, permitindo às classes complementar e até mesmo alterar seu comportamento.
- Interfaces podem ser herdadas em esquema hierárquico, assim como as classes abstratas.

## *Variáveis*

As variáveis guardam informações de um tipo específico.

Podemos, por exemplo, guardar um número inteiro representando o número da conta, um texto para representar o nome do correntista ou um número real para representar o saldo atual da conta. Para utilizar uma variável, devemos primeiramente declará-la no texto do programa.

Na declaração de uma variável, devemos dizer seu tipo (inteiro, texto ou real, por exemplo) e, além disso, qual é o nome que usaremos para referenciá-la no texto do programa.

## Variáveis

Tipo	Tamanho	Valores Possíveis
bool	1 byte	true e false
byte	1 byte	0 a 255
sbyte	1 byte	-128 a 127
short	2 bytes	-32768 a 32767
ushort	2 bytes	0 a 65535
int	4 bytes	-2147483648 a 2147483647
uint	4 bytes	0 to 4294967295
long	8 bytes	-9223372036854775808L to 9223372036854775807L
ulong	8 bytes	0 a 18446744073709551615
float	4 bytes	Números até 10 elevado a 38. Exemplo: 10.0f, 12.5f
double	8 bytes	Números até 10 elevado a 308. Exemplo: 10.0, 12.33
decimal	16 bytes	números com até 28 casas decimais. Exemplo 10.991m, 33.333m
char	2 bytes	Caracteres delimitados por aspas simples. Exemplo: 'a', 'ç', 'o'

# Dúvidas?

---







Dúvidas, críticas e sugestões, entre em contato  
através do e-mail do professor:

[luiz.a.rodriques@cogna.com.br](mailto:luiz.a.rodriques@cogna.com.br)