



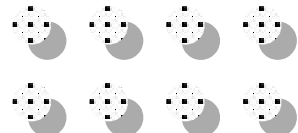
# Programação Orientada a Objetos

Prof. Msc. Rodrigo Nascimento

08/08/25

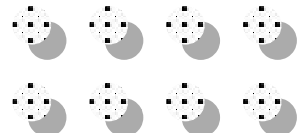
# Sessão da última aula

- Apresentação da Disciplina
  - Ementa;
  - Objetivos Gerais;
  - Metodologia;
  - Avaliação
  - Cronograma das aulas
- Regras e combinados da turma
- APS:
  - APS1;
  - APS2;



# Sessão de hoje

- Contextualização e motivação para POO;
- Definição de POO;
- Elementos básicos da POO;
- Comparação Procedural vs Orientado a Objetos;
- Visão geral dos 4 pilares da POO;
- Demonstração prática;
- Atividades rápidas;
- Resumo de nossa aula;



# Contextualização e motivação para POO – Problemas do paradigma procedural

- O **paradigma procedural** (ou imperativo) organiza o código como uma sequência de **instruções** e **funções** que manipulam dados. Embora tenha sido muito eficaz nas primeiras décadas da computação, ele apresentou limitações à medida que os sistemas se tornaram mais complexos:
  - Baixa modularidade e dificuldade de manutenção;
  - Reuso limitado;
  - Acoplamento excessivo;
  - Escalabilidade prejudicada;
  - Baixa aderência ao domínio do problema;

# Contextualização e motivação para POO – Problemas do paradigma procedural

- Baixa modularidade e dificuldade de manutenção;
  - O código tende a crescer como um “monólito” de funções interdependentes.
- Reuso limitado;
  - Funções podem ser reutilizadas, mas não existe um mecanismo robusto para encapsular dados e comportamentos juntos.
- Acoplamento excessivo;
  - Dados muitas vezes são globais ou compartilhados por várias funções.

# Contextualização e motivação para POO – Problemas do paradigma procedural

- Escalabilidade prejudicada;
  - Projetos grandes se tornam difíceis de gerenciar, pois o paradigma procedural não fornece uma forma natural de representar o mundo real ou organizar responsabilidades de maneira hierárquica.
- Baixa aderência ao domínio do problema;
  - O código procedural descreve mais o “como fazer” do que o “o que é”



# **Por que surgiu a Programação Orientada a Objetos (POO)?**

# **Motivos para o surgimento e adoção da POO**

- Melhor modelagem do mundo real;
- Encapsulamento;
- Reuso e extensibilidade;
- Modularidade e manutenção;
- Escalabilidade;
- Abstração;

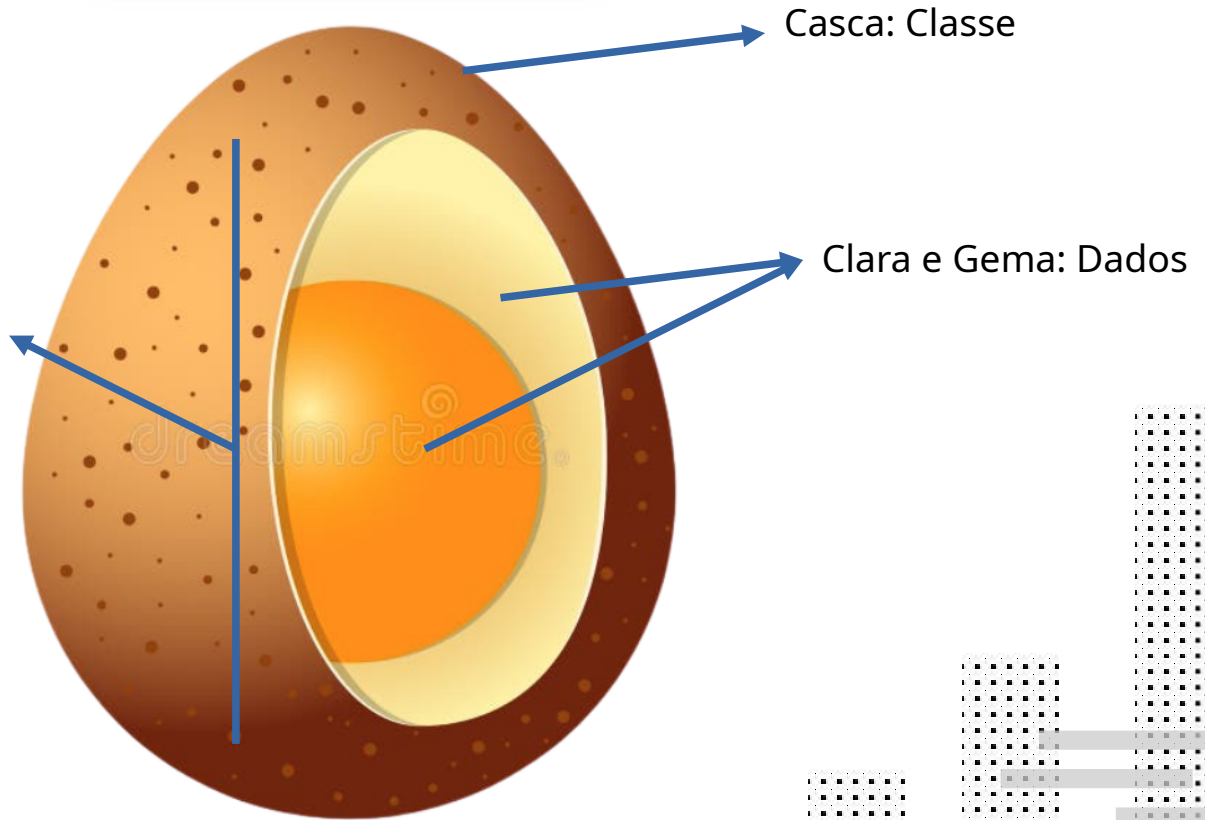


# Melhor modelagem do mundo real



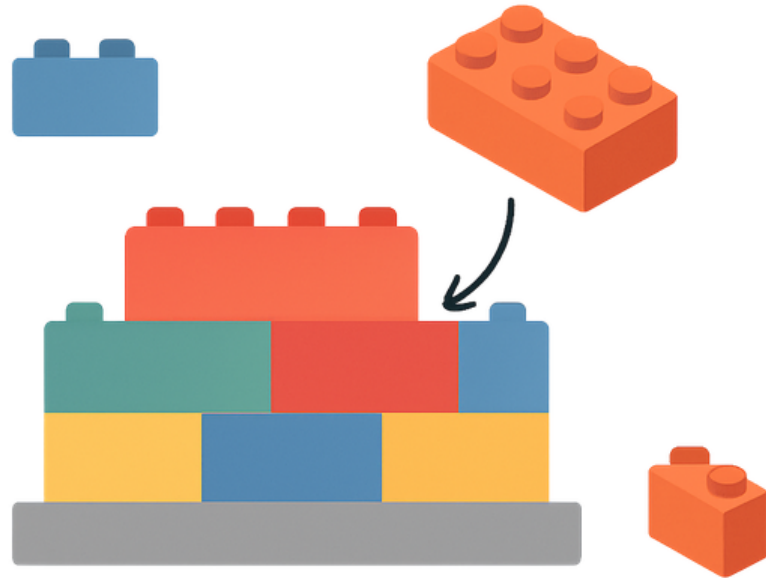
# Encapsulamento

Apenas por uma abertura controlada, é possível acessar o conteúdo.



# Reuso e extensibilidade

Com herança e polimorfismo, é possível criar novas funcionalidades aproveitando código existente, facilitando evolução sem retrabalho excessivo.



# Modularidade e manutenção

Sistemas orientados a objetos podem ser divididos em módulos independentes (classes), o que facilita localizar e corrigir erros sem comprometer o todo.



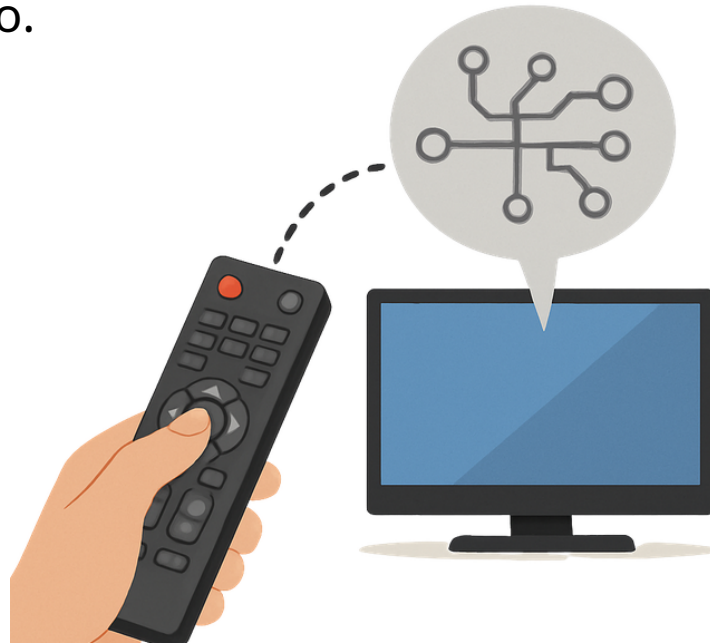
# Escalabilidade

POO oferece um arcabouço conceitual que suporta sistemas grandes, permitindo que múltiplas equipes trabalhem em paralelo com menor risco de conflitos.



# Abstração

Permite focar no que o objeto faz sem precisar se preocupar constantemente com como ele faz, o que facilita a compreensão e colaboração.



# Comparação Procedural vs. Orientado a Objetos



# Atividade Online





# Atividades

- Pesquise sobre os paradigmas de programação e seus tipos? Faça no caderno e vale visto.



# Obrigado!!!

