

# Singleton

Padrão de Projeto de Criação

**INTEGRANTES:** Arthur Monteiro, Brenda Stefany, Bruna Letícia, Elenice Florentina,  
Eduarda Faria e Marco Aurélio de Faria

01

# O problema

- **Cenário:** A necessidade de garantir que uma classe tenha apenas uma única instância durante toda a execução de um sistema.
- **Dificuldade:** Se múltiplas instâncias de uma classe de controle forem criadas, isso pode gerar inconsistência de dados, uso desnecessário de recursos e erros de concorrência.
- **Necessidade:** É crucial para gerenciar o acesso a recursos compartilhados, como um arquivo de log, um objeto de configuração global ou um pool de conexões com o banco de dados.

02

## A solução

- **Abordagem:** O padrão garante que a própria classe seja responsável por controlar sua criação e garantir sua singularidade.
- **Mecanismo:**
  - O construtor da classe é definido como privado, o que impede que ela seja instanciada diretamente com o operador new.
  - A classe mantém uma instância estática e privada de si mesma.
  - A classe fornece um método público e estático que retorna sempre essa mesma instância única.

# 03

## Estrutura

- O padrão é implementado em uma única classe, que contém:
  - **Instância Estática Privada:** Atributo que armazena a instância única da classe.
  - **Construtor Privado:** Método que impede a instanciação por qualquer outra classe.
  - **Método Estático de Acesso Global:** Ponto de acesso público que retorna a instância única, criando-a apenas na primeira chamada.

# 04

## Vantagens

- Singularidade
- Acesso Global
- Criação Sob Demanda

## Desvantagens

- Introduz estado global
- Dificulta Testes
- Risco com Threads

# 05

## Aplicabilidade do Padrão

- É estritamente necessário que uma classe tenha uma, e somente uma, instância.
- Você precisa de um ponto de acesso global e unificado para essa instância.
- É preciso controlar o acesso a recursos compartilhados e únicos no sistema.

# Conclusão

- O Singleton é o padrão ideal para centralizar o controle de recursos, garantindo que uma classe tenha uma instância única.
- Ele dá à própria classe o poder de gerenciar sua existência e oferece um ponto de acesso compartilhado para toda a aplicação.
- Deve ser utilizado com moderação, pois o uso excessivo pode levar a um forte acoplamento e dificultar a manutenção e os testes do código.