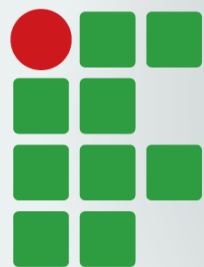


AULA 3 – Padrão de Projetos

Prof. Igor Justino Rodrigues
25/10/2024





**INSTITUTO
FEDERAL**

Goiano

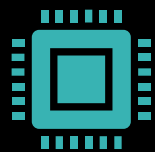
Campus
Ceres

OBJETIVOS DA AULA

- Explicar a origem e o propósito dos padrões de projeto, destacando a importância da reutilização de soluções consagradas.
- Descrever o funcionamento do padrão Singleton e suas principais características, como a garantia de uma única instância e o controle de acesso global.
- Descrever o funcionamento do padrão Factory, explicando sua lógica de criação de objetos e como ele promove o desacoplamento entre a lógica de criação e a aplicação.
- Demonstrar a aplicação prática dos padrões Singleton e Factory em código, destacando suas vantagens e limitações.

17/10/2024

Origem dos Padrões de Projeto



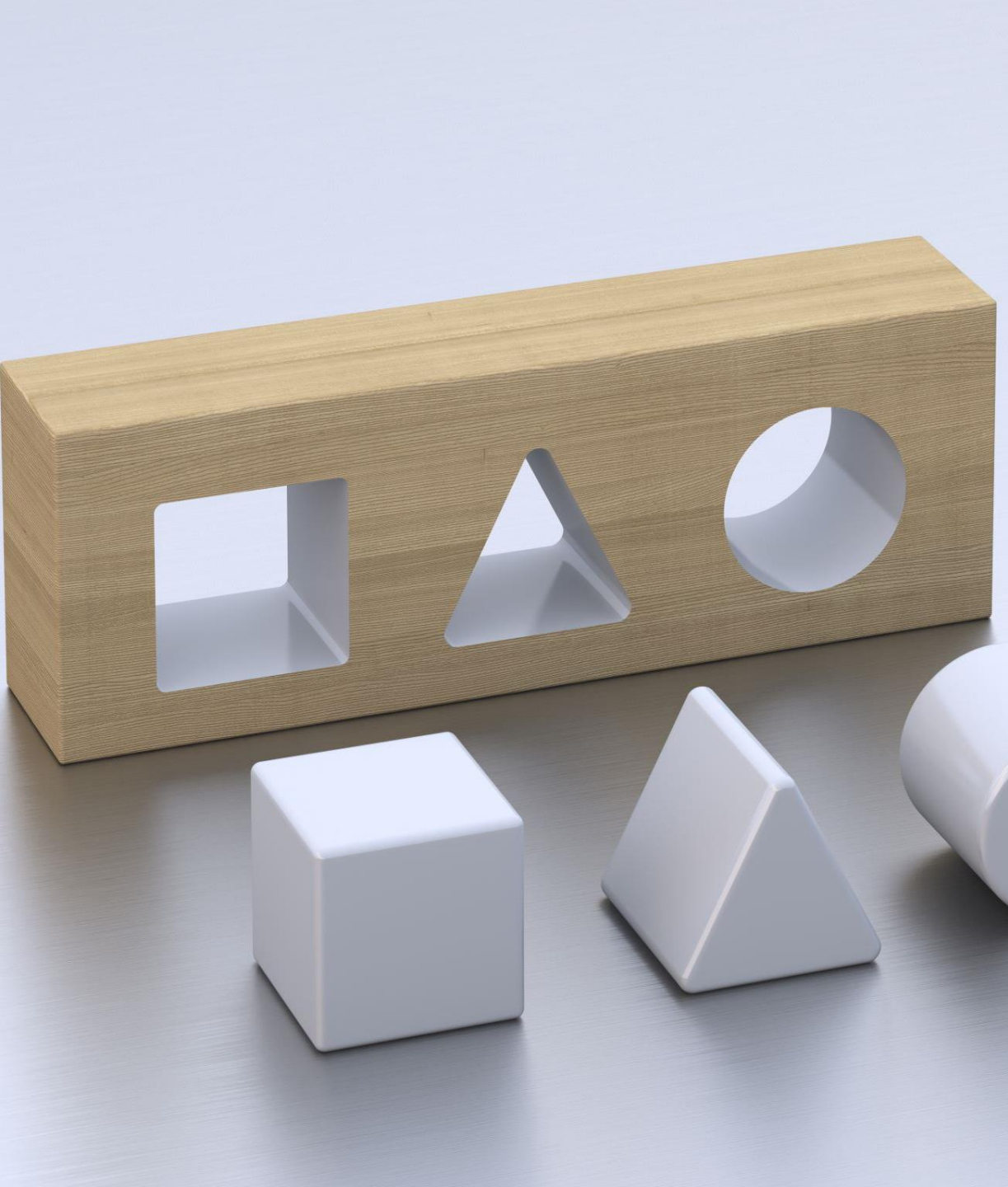
Surgiram na década de 1990 com o livro *"Padrões de Projeto: Elementos de Software Orientado a Objetos Reutilizáveis"*



Objetivo: Padronizar soluções recorrentes para problemas de design de software.



Importância: Reutilização de boas práticas e soluções consagradas.

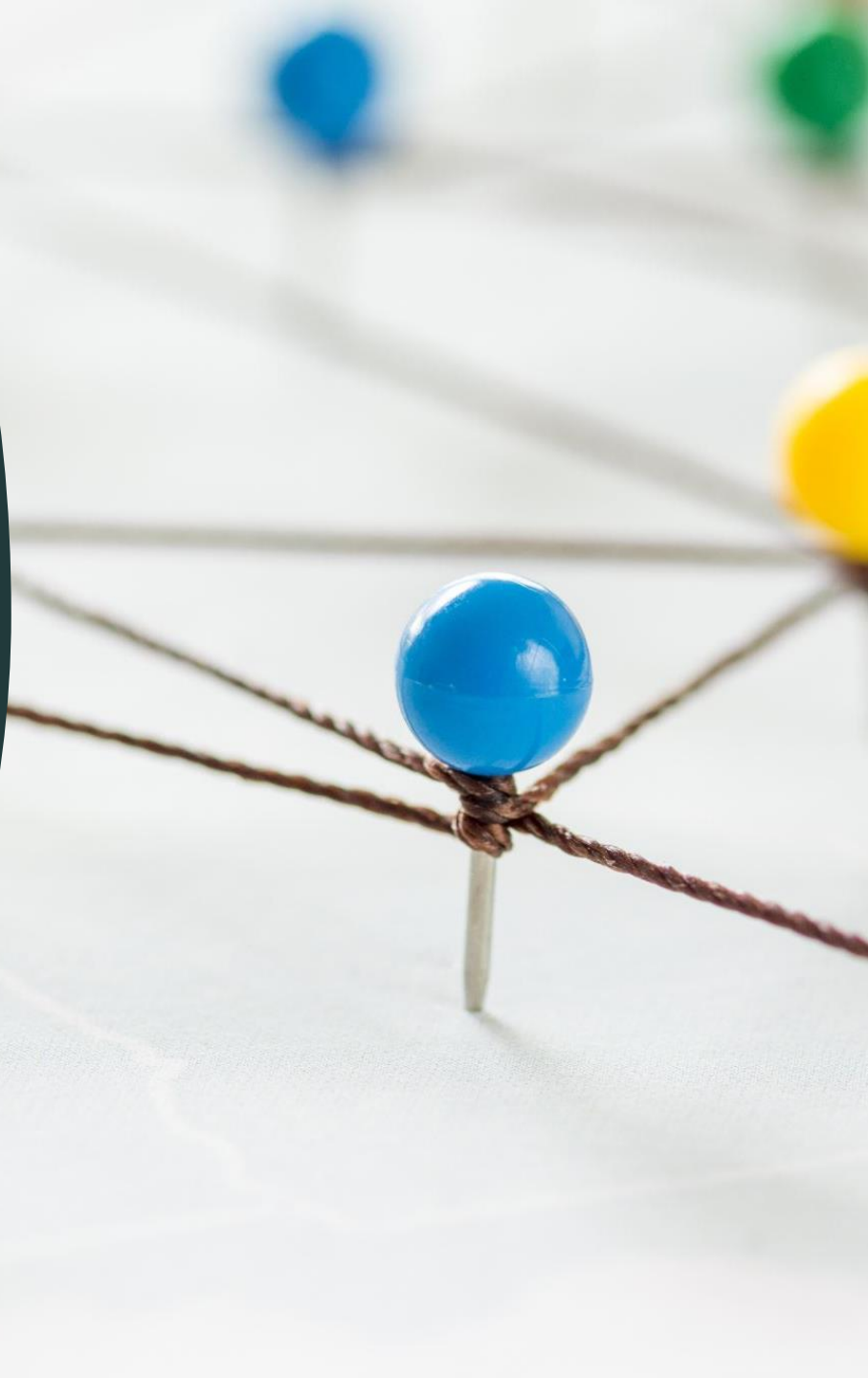


Padrões de Criação

- Foco na **instanciação de objetos** de forma controlada.
- **Problema comum:** Como criar objetos de forma flexível, eficiente e com menor acoplamento.
- **Exemplos de padrões de criação:** Singleton, Factory, Builder, Prototype.

Padrão Singleton

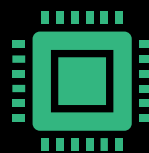
- **Conceito:** Garante que uma classe tenha apenas uma instância e fornece um ponto de acesso global a essa instância.
- **Objetivo:** Controlar e otimizar o uso de recursos globais, como conexões de banco de dados ou gerenciadores de logs.
- **Características:**
 - Instância única.
 - Construtor privado.
 - Método público e estático para retornar a instância.



Aplicações e Benefícios do Singleton



Usado em sistemas onde é necessário um controle centralizado (gerenciadores de configuração, conexão de banco de dados).

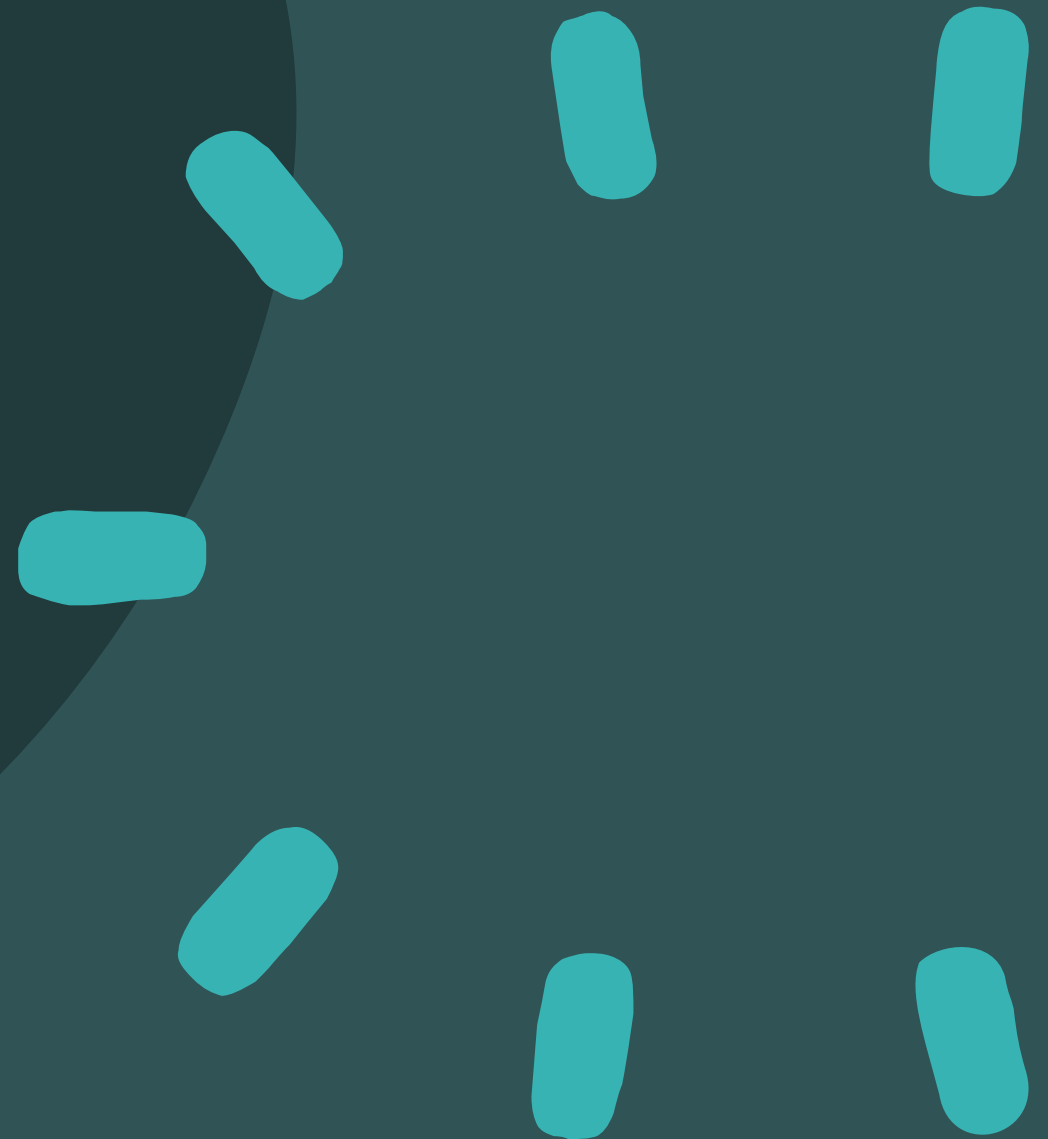


Benefícios: Economia de memória, controle de acesso, garantia de unicidade.



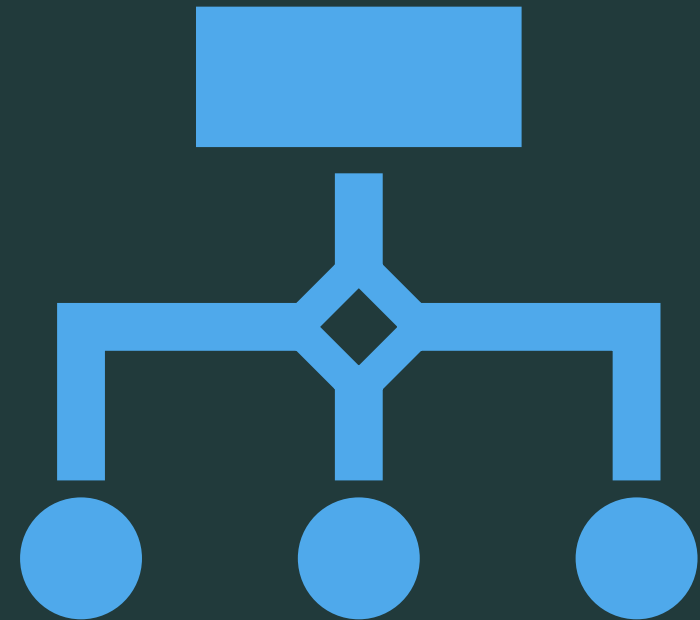
Atenção: Pode levar a problemas de acoplamento excessivo e dificultar testes unitários.

Exemplos

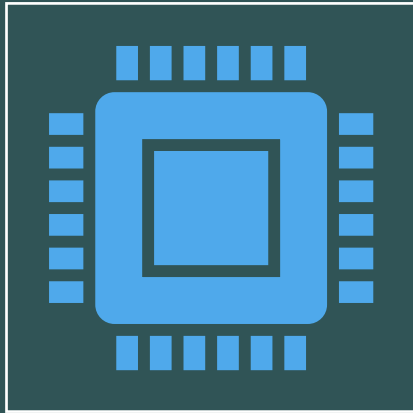


Padrão Factory

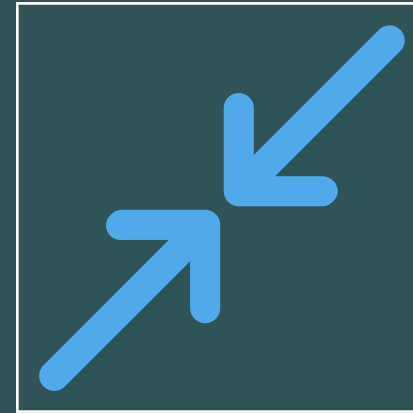
- **Conceito:** Define uma interface para criar objetos, mas permite que subclasses decidam quais classes instanciar.
- **Objetivo:** Desacoplar a criação de objetos da lógica de negócio.
- **Características:**
 - Flexibilidade na criação de diferentes tipos de objetos.
 - Uso de classes abstratas ou interfaces.
 - Centralização da lógica de criação.



Aplicações e Benefícios do Factory



Usado em sistemas onde a lógica de criação de objetos varia (sistemas com múltiplos tipos de usuários ou veículos).



Benefícios: Flexibilidade, desacoplamento da criação e menor impacto ao adicionar novos tipos de objetos.

The image features six teal, hand-drawn brushstroke shapes arranged in two vertical columns of three on either side of the central text. The shapes are irregular and elongated, resembling thick, expressive strokes.

Exemplos

Singleton vs Factory



Singleton: Garantia de uma instância única para controle centralizado.



Factory: Flexibilidade na criação de objetos sem acoplar a lógica de criação à classe de negócios.



Semelhanças: Ambos ajudam a gerenciar e controlar a criação de objetos.



Diferenças: Singleton foca na instância única; Factory na flexibilidade de criação.