

**Nome:** João Kennedy Souza Soares

**Curso:** Ciência da Computação

**Disciplina:** POO

## **Seminário Bridge**

### **1. Introdução**

O presente relatório aborda a implementação do padrão de projeto Bridge em Java, utilizando como exemplo um sistema que relaciona formas geométricas (quadrado, círculo) com diferentes cores (azul, vermelho). O padrão Bridge visa separar a abstração da implementação, permitindo que ambas possam variar independentemente.

### **2. Categorias e Objetivos dos Padrões de Projeto**

#### **2.1 Contextualização e Motivação**

O padrão Bridge surgiu da necessidade de desacoplar uma abstração de sua implementação, proporcionando maior flexibilidade e extensibilidade no código. Ao separar esses componentes, o sistema torna-se mais adaptável a mudanças, facilitando a introdução de novas funcionalidades ou a alteração de comportamentos existentes.

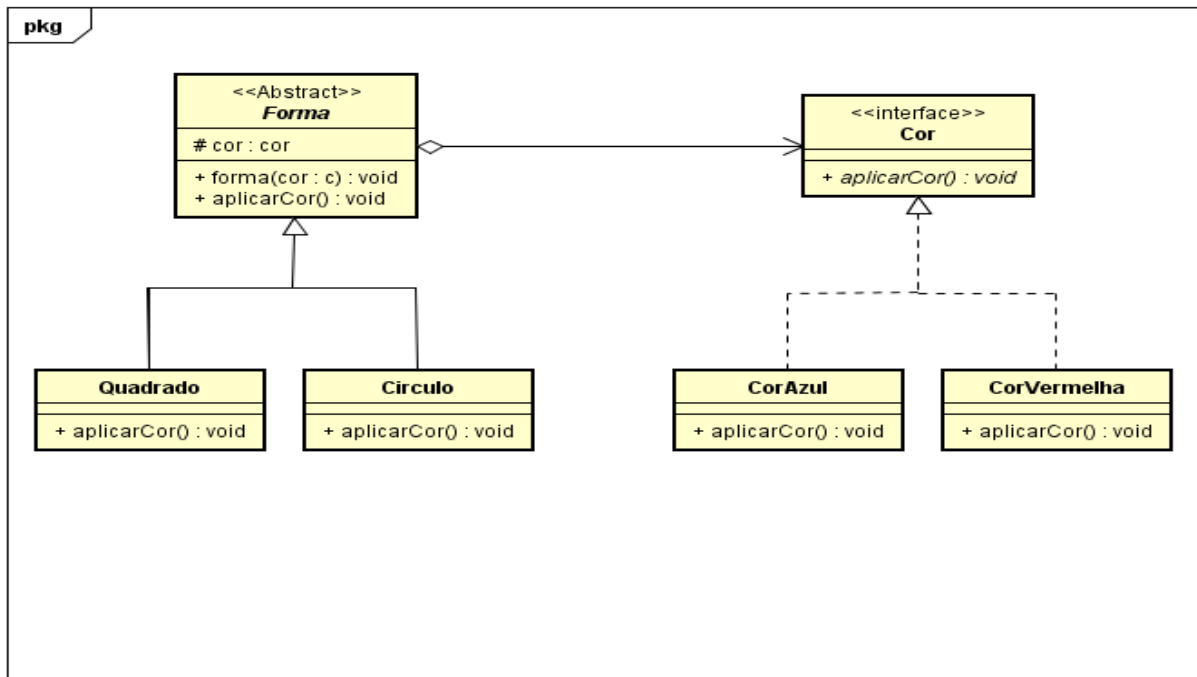
#### **2.2 Categorias de Padrões de Projeto**

Os padrões de projeto podem ser categorizados em três grupos principais: criacionais, estruturais e comportamentais. O padrão Bridge se enquadra na categoria estrutural, que trata da composição de classes e objetos.

### **3. Objetivo, Problema Resolvido, Conceitos, Funcionamento e Aplicação**

O padrão Bridge é uma abordagem eficaz para desacoplar a abstração de sua implementação em sistemas de software. Seu principal objetivo é permitir que essas duas partes evoluam independentemente, promovendo flexibilidade e facilitando a extensão do sistema. Ao separar a hierarquia de classes em uma parte abstrata e outra de implementação, o Bridge oferece uma ponte flexível entre esses dois elementos, permitindo que diferentes abstrações possam ser combinadas de maneira versátil com implementações específicas em tempo de execução. Essa flexibilidade torna o padrão Bridge aplicável em uma variedade de contextos, desde interfaces gráficas até sistemas de gerenciamento de banco de dados e drivers de dispositivos, proporcionando uma arquitetura mais modular e fácil de manter.

### **4. Diagrama de Classe UML**



Fonte: astah community

## 5. Códigos-fonte

Link github: [https://github.com/JoaoKSS/Poo\\_Bridge.git](https://github.com/JoaoKSS/Poo_Bridge.git)

## Conclusão

A implementação do padrão Bridge neste sistema de formas geométricas e cores oferece uma visão clara de como a abstração e a implementação podem evoluir independentemente. Ao desacoplar esses componentes, tornamos o sistema mais flexível, facilitando a introdução de novas formas e cores sem afetar a estrutura existente. O padrão Bridge, portanto, revela-se uma escolha valiosa para sistemas onde a hierarquia de abstrações e implementações precisa de flexibilidade.

## Referências

Baeldung. Java Bridge Pattern. Baeldung, 2023-08-23. Disponível em: <https://www.baeldung.com/java-bridge-pattern>. Acesso em: 21 nov. 2023.

DigitalOcean Community. Bridge Design Pattern in Java. DigitalOcean Community, 2023-01-25. Disponível em: <https://www.digitalocean.com/community/tutorials/bridge-design-pattern-java>. Acesso em: 21 nov. 2023.

Jones Roberto. Design Patterns - Parte 9 - Bridge. Medium, 2023-06-08. Disponível em: <https://medium.com/@jonesroberto/desing-patterns-parte-9-bridge-5ca127f72de>. Acesso em: 21 nov. 2023.