

A6- Aplicando Conhecimento

COMPONENTE CURRICULAR:	DESENVOLVIMENTO DE SISTEMAS II
NOME DO ALUNO:	JOÃO PEDRO LIMA LUSTOSA AMORIM
RA:	10289920

1. Definição e Objetivo dos Padrões GoF Estruturais:

- a. O que são os padrões GoF estruturais e qual é seu objetivo principal no desenvolvimento de software orientado a objetos?

R: Os Padrões GoF (*Gang of Four*) são um conjunto de soluções para problemas comuns na organização de classes e objetos, com a finalidade de definir como as classes e os objetos se relacionam, colaborando para que o sistema seja mais organizado e flexível, facilitando assim na manutenção.

2. Padrão Bridge:

- a. Qual é o problema que o padrão Bridge pretende resolver?

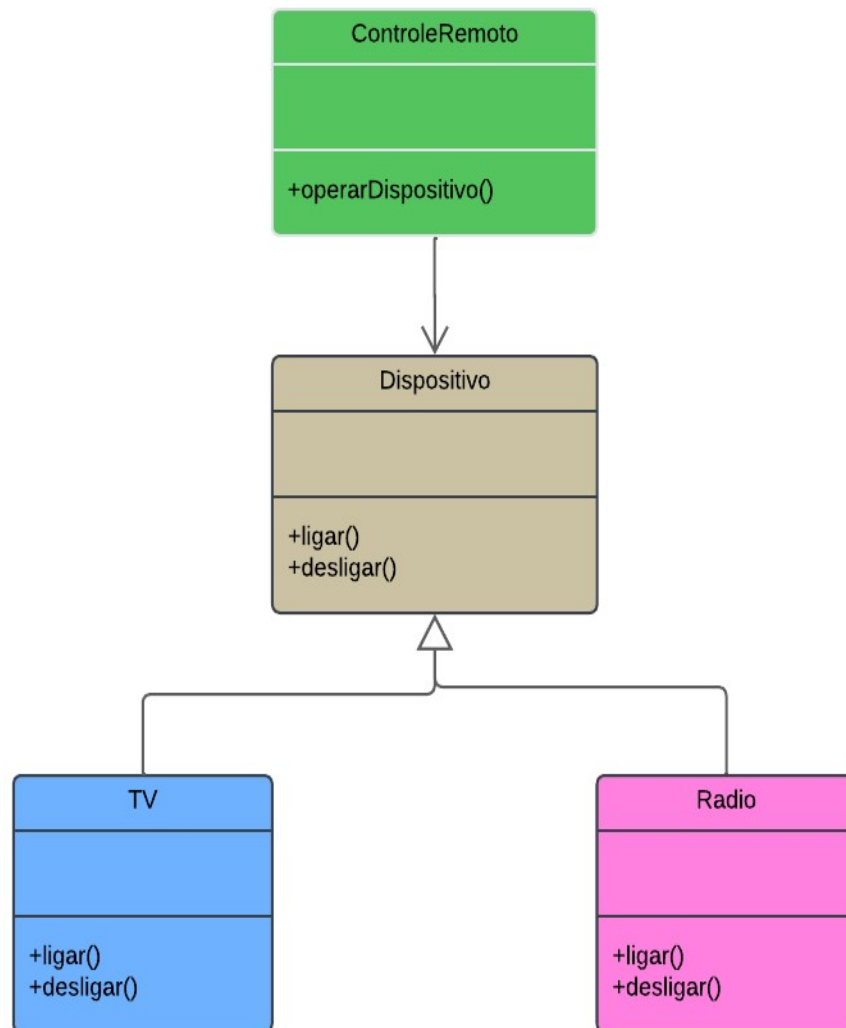
R: O Padrão Bridge tenta solucionar o problema de quando temos uma hierarquia de classes muito complexa com excesso de subclasses que representam variações de funcionalidades ou implementações, dificultando o aprimoramento do código por expansão. O Padrão Bridge separa cada uma abstração da sua implementação, facilitando a criação de novas classes.

- b. Explique como a solução Bridge é implementada.

R: A solução Bridge separa a abstração e a implementação em duas hierarquias separadas referenciadas, podendo cada lado ser estendido independentemente sem afetar o outro lado.

- c. Descreva um exemplo do mundo real onde o padrão Bridge pode ser aplicado e faça a modelagem UML desse exemplo.

R: Um exemplo do Padrão Bridge são os controles remotos universais (como por exemplo os celulares que contém aplicativos para controlar televisões, ar-condicionado etc.). Para evitar criar uma classe para cada combinação possível de dispositivo e controle remoto, o padrão Bridge separa a abstração (o controle remoto) da implementação (os dispositivos), permitindo ao controle controlar qualquer dispositivo sem depender de um controle remoto específico.



3. Padrão Decorator:

a. Qual é o problema que o padrão Decorator pretende resolver?

R: O padrão Decorator soluciona o problema da adição de funcionalidades a objetos de maneira flexível, sem precisar modificar as classes originais ou criar muitas subclasses. É útil quando queremos adicionar funcionalidades extras em tempo de execução, sem afetar outros objetos da mesma classe.

b. Descreva a estrutura básica do padrão Decorator.

R: Na estrutura básica do Decorator, temos uma interface comum para o objeto principal e para o Decorator, que contém o objeto que será decorado. O Decorator envolve o objeto original, adicionando comportamento antes ou depois de chamar os métodos do objeto decorado.

c. Dê um exemplo de aplicação do padrão Decorator em uma situação do mundo real e faça a modelagem UML desse exemplo.

R: Temos o sistema de pedidos de uma cafeteria na qual os clientes podem personalizar suas bebidas. O café pode receber adicional de leite, chocolate, caramelo, com um acréscimo no valor do pedido. A classe base será Bebida e Cafe é uma classe concreta que implementa Bebida definindo o custo básico de uma bebida simples. O Decorator será BebidaDecorator, uma classe abstrata que referencia Bebida. LeiteDecorator e ChocolateDecorator são os decoradores concretos que estendem BebidaDecorator, adicionando um acréscimo ao valor total da bebida

