

Relatório Técnico: Padrão de Projeto Facade

1. Descrição do Padrão

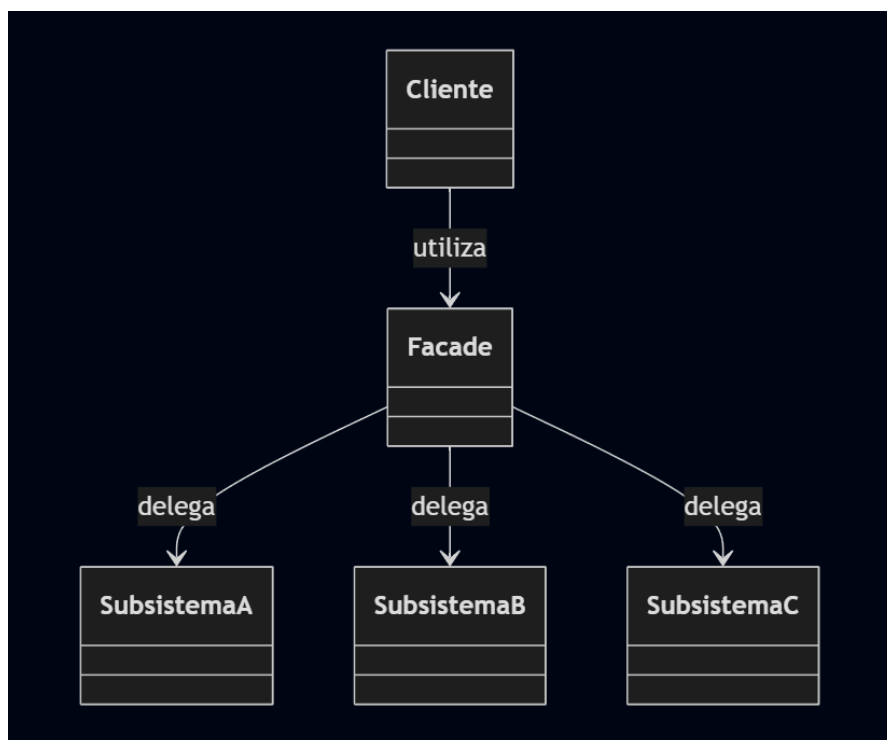
O **Facade** é um padrão de projeto estrutural que fornece uma interface simplificada para uma biblioteca, um framework ou qualquer outro conjunto complexo de classes.

- **Objetivo:** O principal objetivo do padrão Facade é fornecer uma interface de alto nível que torna um subsistema ou um conjunto de interfaces mais fácil de usar. Ele envolve uma única classe que representa todo um subsistema, ocultando sua complexidade dos clientes.
- **Problema:** Sistemas complexos frequentemente possuem múltiplas classes interdependentes. Para um cliente realizar uma tarefa, ele pode precisar interagir com diversos objetos desse sistema, conhecendo em detalhes a lógica e a ordem correta das operações. Isso gera um alto acoplamento entre o cliente e o subsistema, tornando o código do cliente difícil de entender, manter e evoluir. O Facade resolve esse problema ao introduzir uma camada de abstração que desacopla o cliente da complexidade do subsistema.

2. Implementação do Padrão

A implementação do Facade envolve a criação de uma nova classe, a **Facade**, que encapsula a complexidade do subsistema. O cliente interage apenas com essa facade, que por sua vez delega as chamadas para os objetos apropriados dentro do subsistema.

Estrutura:



- **Facade:**
 - Conhece quais classes do subsistema são responsáveis por uma requisição.
 - Delega as requisições do cliente para os objetos apropriados do subsistema.
 - Não encapsula totalmente o subsistema. Os clientes podem, se necessário, acessar as classes do subsistema diretamente.
 - **Subsistema:**
 - Implementa a funcionalidade complexa.
 - Realiza o trabalho real solicitado pela Facade.
 - Não tem conhecimento da existência da Facade; ou seja, a Facade não adiciona nenhuma referência a eles.
 - **Cliente:**
 - Utiliza a classe Facade para interagir com o subsistema de forma simplificada.
-

3. Cenário de Aplicação do Padrão

Cenário: Sistema de Home Theater

Vamos imaginar um sistema de Home Theater composto por vários equipamentos: uma televisão, um sistema de som, um reproduutor de Blu-ray, luzes ambiente e uma pipoqueira.

Problema: Para preparar uma sessão de cinema em casa, um usuário precisa executar uma série de passos em uma ordem específica:

1. Ligar a pipoqueira e preparar a pipoca.
2. Diminuir a intensidade das luzes ambiente.
3. Ligar a televisão.
4. Configurar a televisão para a entrada de vídeo correta.
5. Ligar o amplificador.
6. Configurar o som para o modo "Ambiente".
7. Ajustar o volume do som.
8. Ligar o reproduutor de Blu-ray.
9. Inserir e iniciar a reprodução do filme no Blu-ray.

Realizar todas essas ações manualmente é complexo e tedioso. Se o cliente precisar interagir com cada classe de dispositivo, o código se tornará complexo e fortemente acoplado a cada um desses componentes.

Aplicação do Padrão Facade:

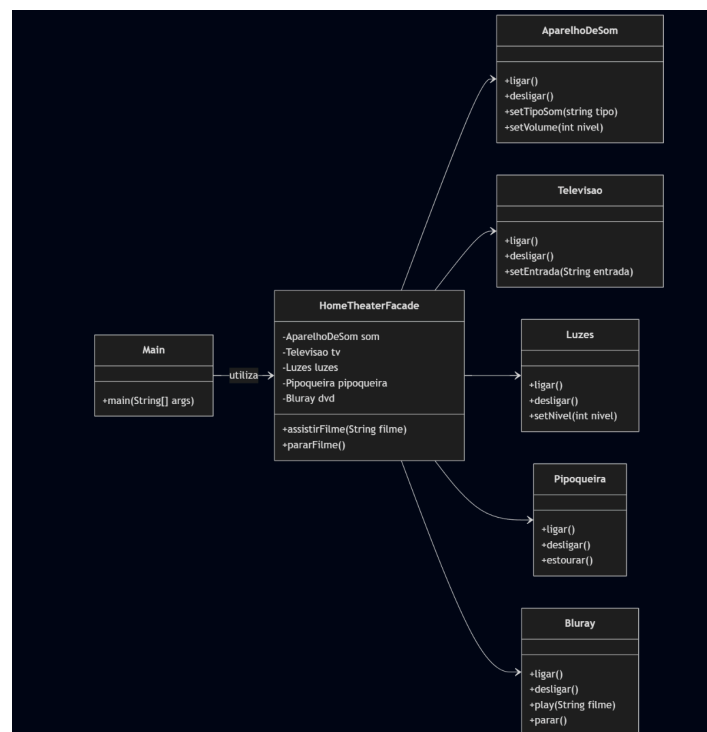
Para simplificar, criamos uma classe `HomeTheaterFacade`. Esta facade fornecerá métodos simples e de alta funcionalidade, como `assistirFilme(String filme)` e `pararFilme()`.

- Ao chamar `assistirFilme("O Senhor dos Anéis")`, a facade executará internamente toda a sequência de operações descrita acima, interagindo com cada um dos componentes do subsistema na ordem correta.
- Ao chamar `pararFilme()`, a facade executará a sequência inversa para desligar tudo de forma organizada.

Dessa forma, o cliente só precisa conhecer e interagir com a `HomeTheaterFacade`, que oferece uma interface limpa e simples para uma operação complexa.

4. Diagrama de Classes da Solução

A seguir, o diagrama de classes da solução implementada para o cenário do Home Theater.



Explicação do Diagrama:

- A classe **Main** (cliente) interage unicamente com a **HomeTheaterFacade**.
- A **HomeTheaterFacade** possui referências a todos os objetos do subsistema: **Aparelho de som**, **Televisao**, **Luzes**, **Pipoqueira** e **BluRay**.
- Quando o **Cliente** chama um método como `assistirFilme()`, a **HomeTheaterFacade** delega as chamadas para os métodos apropriados dos objetos do subsistema (`luzes.setNivel(10)`, `tv.ligar()`, `player.play()`, etc.), orquestrando a operação completa.
- As classes do subsistema (**Aparelho de som**, **Televisao**, etc.) não sabem da existência da **HomeTheaterFacade**. Elas simplesmente expõem suas funcionalidades.