

Instituto Federal
Campus Goiânia

Bacharelado em Sistemas de Informação

POO II

Prof. Dory Gonzaga Rodrigues



Ementa

Padrões de Projeto





Padrões de Projeto

Padrões

Soluções reutilizáveis para situações ou problemas encontrados comumente em desenvolvimento de software orientado a objetos.

Livros

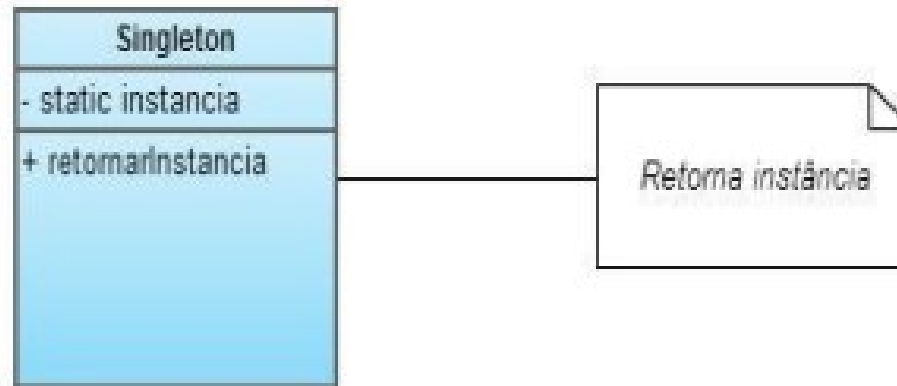
- Design Patterns: Elements of Reusable ObjectOriented Software (Gangue dos Quatro - Gang of Four ou GoF)
- Core J2EE Patterns

Padrões de Projeto

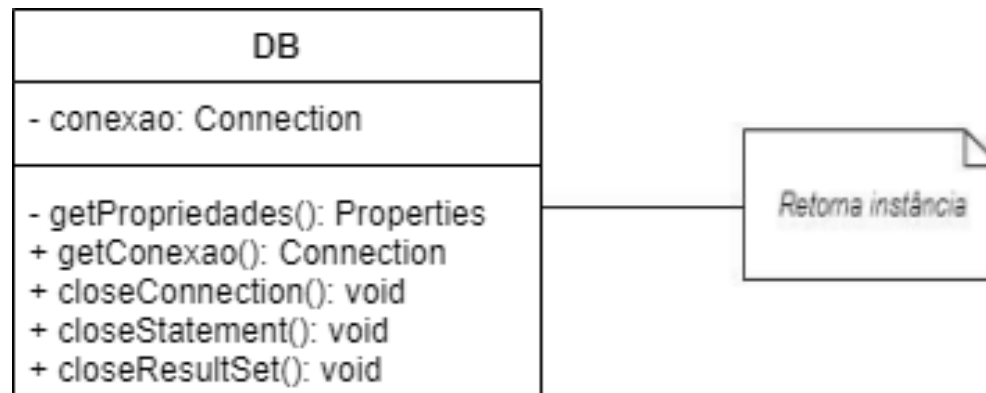
SINGLETON

Objetivo: Criar apenas um objeto de uma determinada classe.

Características: Reutilizar objetos



Exemplo:



Padrões de Projeto

SINGLETON

```
public static Connection getConexao() {  
    String host;  
    String user;  
    String password;  
  
    try {  
  
        Properties p = DB.getProperties();  
        if (conexao == null) {  
            host = p.getProperty("host");  
            user = p.getProperty("user");  
            password = p.getProperty("password");  
            conexao = DriverManager.getConnection(host, user, password);  
        }  
  
    } catch (SQLException | IOException e) {  
        e.printStackTrace();  
    }  
  
    return conexao;  
}
```

Observe o código do método getConexao:

- Ele somente irá criar uma instância da conexão se ainda não existir (null).
- Depois, sempre que o método for chamado, ele retornará a instância do objeto já existente.

```
public static void main(String[] args) {  
  
    CursoDAO cursoDao = new CursoDAOImp(DB.getConexao());  
  
    Curso c1 = new Curso(null, "POO II");  
    cursoDao.insert(c1);  
}
```



Padrões de Projeto

DAO (Data Access Object)

Objetivo: Encapsular o acesso a dados em uma classe separada da aplicação. Ou seja, manipula os dados em uma camada lógica separada, gerenciando a conexão com a fonte de dados para obter e armazenar os dados.

O padrão DAO em conjunto com padrões de projeto que atuam como fábricas de objetos (Factory e Abstract Factory) possibilita a implementação de acesso para diferentes mecanismos de persistência.



Padrões de Projeto

DAO (Data Access Object)

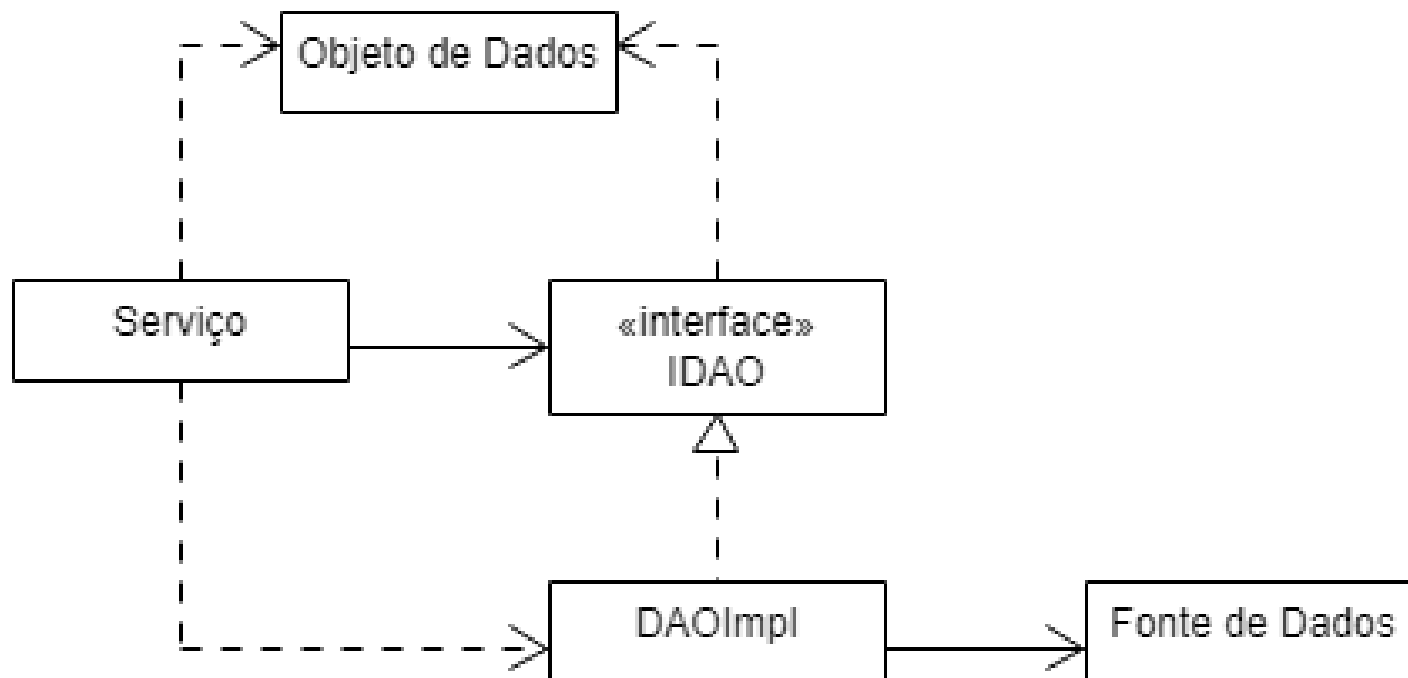
Características

- Centralização do código de acesso/manipulação de dados da aplicação.
- Separação da lógica de negócio da persistência.
- Tornar transparente o acesso aos dados nas aplicações.
- Possibilitar acesso a diferentes fontes de dados de forma transparente para o usuário.



Padrões de Projeto

DAO (Data Access Object)





Padrões de Projeto

DAO (Data Access Object)

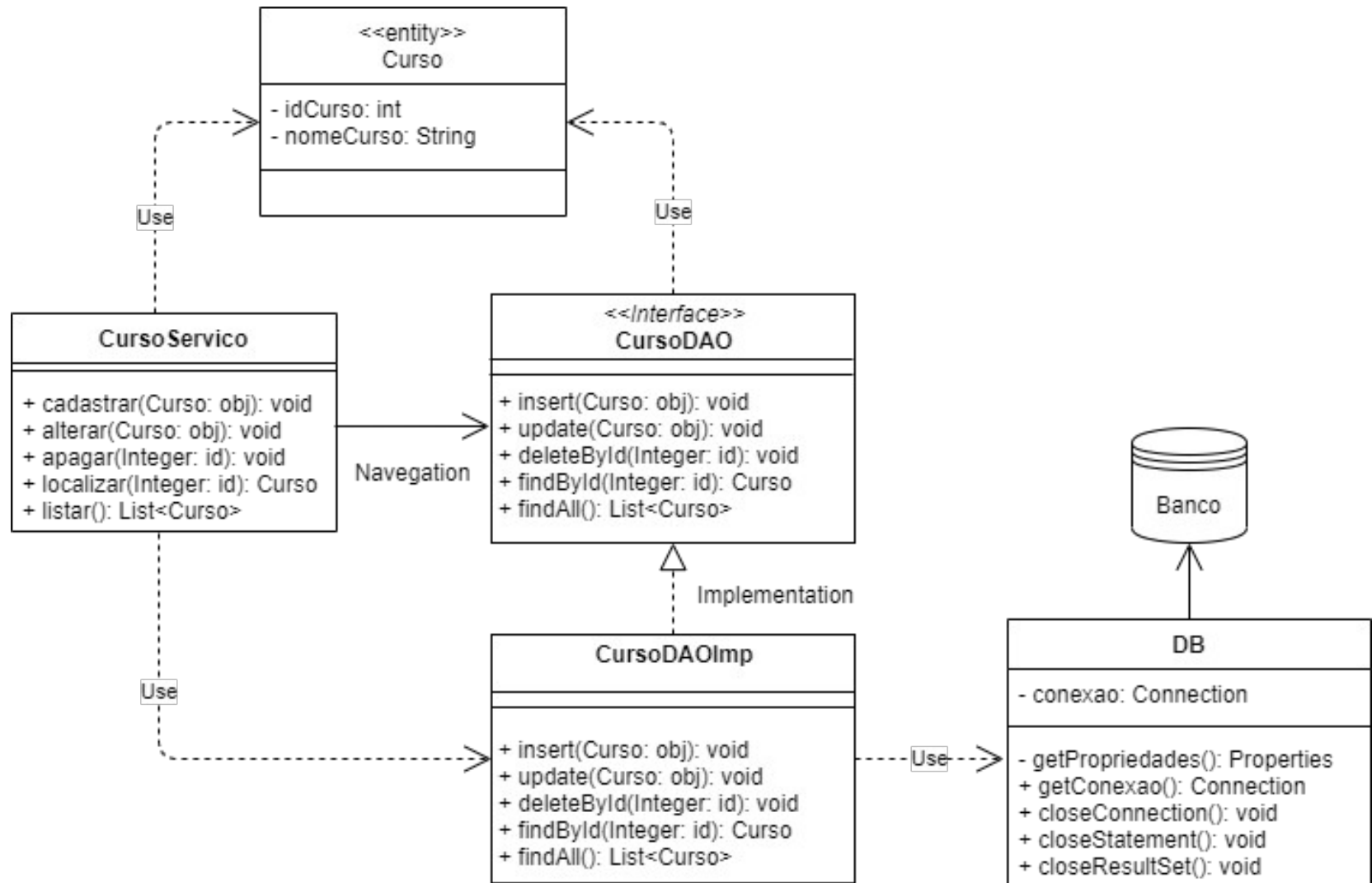
Interface DAO: Essa interface define as operações padrão a serem executadas em um objeto de dados

Classe Implementa DAO: Esta classe implementa a Interface DAO. Essa classe é responsável por obter dados de uma fonte de dados que pode ser de um banco de dados, XML ou qualquer outro mecanismo de armazenamento.

Objeto de Dados (ou de Modelo/Domínio/Entidade): Este é um objeto que contém métodos get/set para armazenar dados recuperados usando a classe DAO.

Padrões de Projeto

DAO (Data Access Object)





Padrões de Projeto

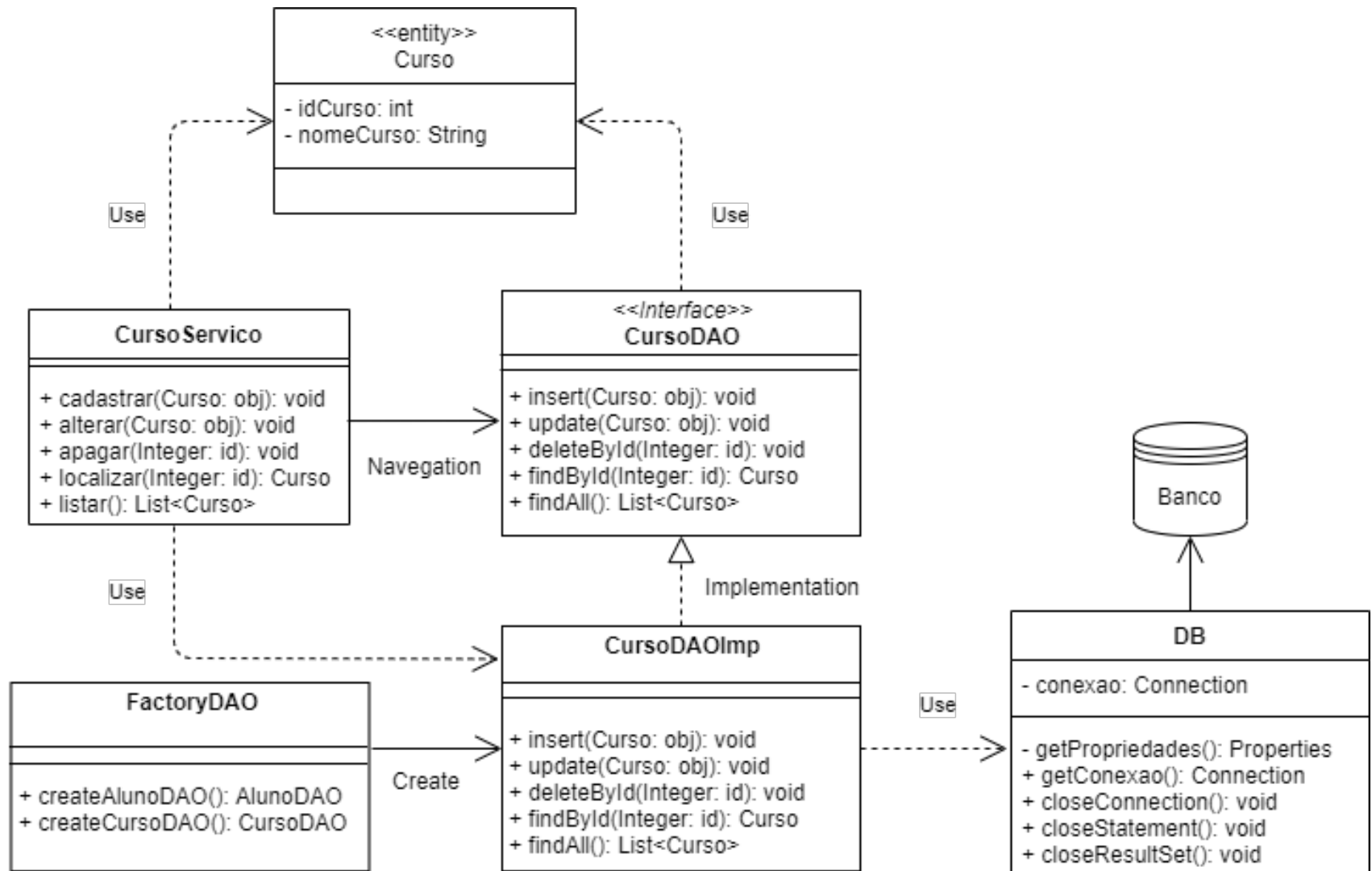
Factory Method

Objetivo: definir uma interface para criar um objeto mas deixar que subclasses decidam que classe instanciar.

Permite que uma classe delegue a responsabilidade de instanciamento às subclasses.

Padrões de Projeto

Factory Method





Padrões de Projeto

Modelo MVC

Padrão de desenvolvimento de software baseado em 03 camadas:

- Modelo (Model)
- Visão (View)
- Controladora (Controller)

Objetivo: separar a lógica da aplicação da apresentação das informações ao usuário.



Padrões de Projeto

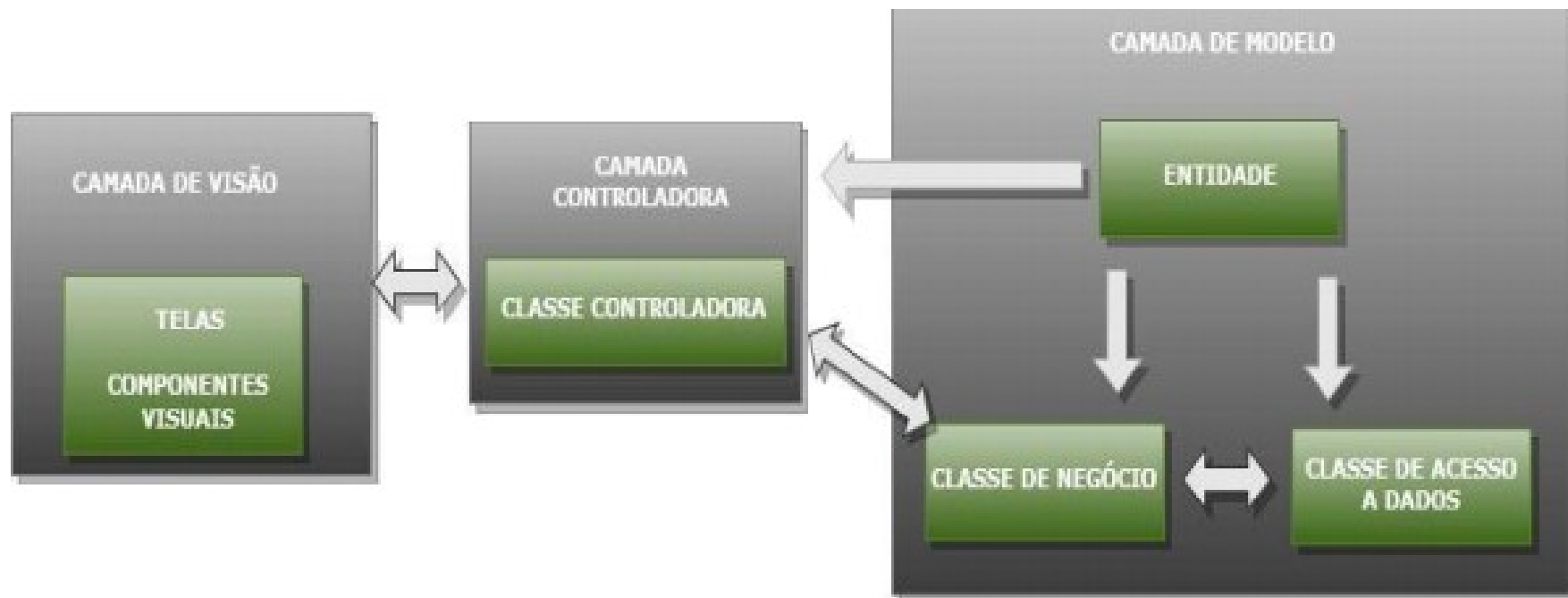
Modelo MVC

- **Modelo (Model)**
concentra as classes de domínio (entidades) da aplicação, além das classes de negócio e de acesso a dados.
- **Visão (View)**
responsável pelo layout da aplicação (telas em HTML, por exemplo) e seus componentes visuais;
- **Controladora (Controller)**
direciona o fluxo de dados entre as camadas de visão e de modelo da aplicação.



Padrões de Projeto

Modelo MVC

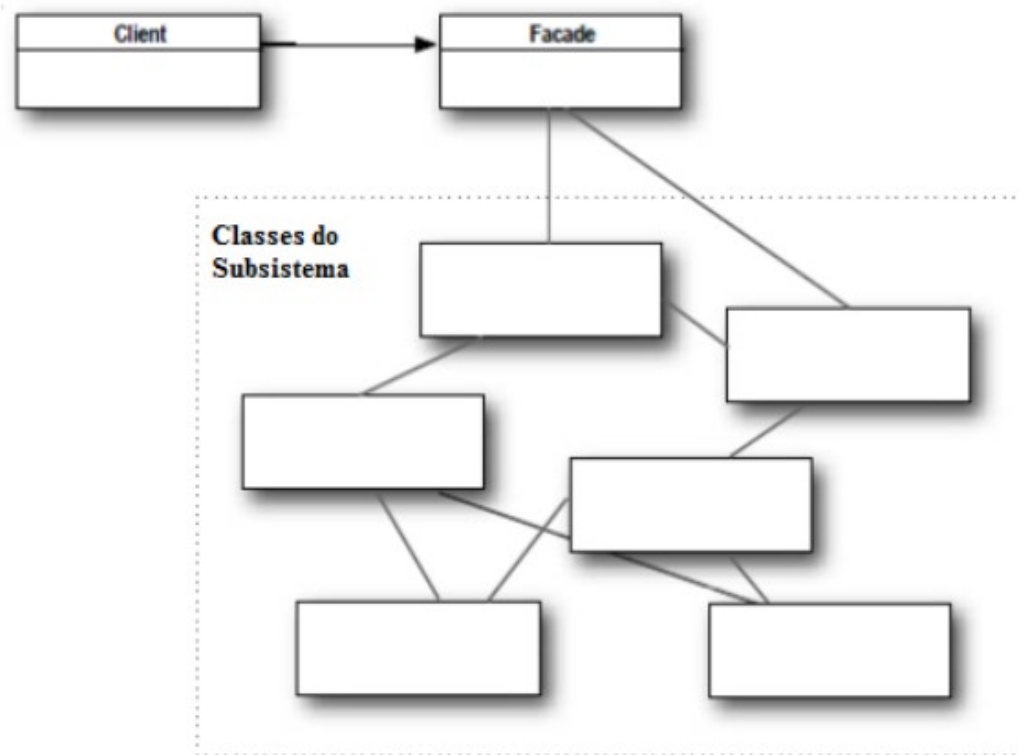




Padrões de Projeto

Facade

Objetivo: ocultar a complexidade de uma ou mais classes, simplificando o uso de um subsistema implementando apenas uma classe que forneça uma interface única e mais razoável.





Padrões de Projeto

Facade

Características:

- fornece uma interface unificada para um conjunto de interfaces em um subsistema.
- define uma interface de nível mais alto que facilita a utilização do subsistema
- Um subsistema pode ter diversos *Facades*.

Padrões de Projeto

Modelo MVC com Facade

