

# Planejamento de Padrões de Projeto (GoF) para o Planetário Virtual

Este documento consolida as sugestões de padrões de projeto GoF e a proposta de distribuição em equipes para o desenvolvimento do projeto do planetário virtual.

## 1. Sugestão de Padrões de Projeto (GoF) por Categoria

A seguir, são apresentados os padrões de projeto GoF selecionados, divididos por suas categorias, com uma breve justificativa de aplicação no contexto do projeto.

### Padrões Criacionais (Creational)

Padrões que abstraem o processo de instanciação de objetos.

#### 1. Builder

- **Onde Aplicar:** Na criação de objetos da classe Usuario.
- **Justificativa:** A classe Usuario possui diversos atributos, alguns opcionais (sobre, fotoUrl). O Builder permite uma construção passo a passo do objeto, tornando o código mais legível e flexível para lidar com múltiplas configurações de um usuário.

#### 2. Factory Method

- **Onde Aplicar:** Na criação de objetos Postagem e Comentário.
- **Justificativa:** Define uma interface para criar um objeto, mas deixa as subclasses decidirem qual classe instanciar. Pode-se ter uma FabricaDeConteudo com métodos como criarPostagem() e criarComentario(), permitindo que o sistema trate diferentes tipos de conteúdo de forma coesa e facilitando a adição de novos tipos de conteúdo no futuro.

#### 3. Singleton

- **Onde Aplicar:** Para gerenciar o acesso aos dados dos planetas (ex: ServicoPlanetario).
- **Justificativa:** Garante que uma classe tenha apenas uma instância e fornece um ponto global de acesso a ela. Ideal para gerenciar recursos compartilhados, como dados de planetas que são carregados uma vez e consultados por várias partes da aplicação.

### Padrões Estruturais (Structural)

Padrões que lidam com a composição de classes e objetos para formar estruturas maiores e mais flexíveis.

#### 1. Decorator

- **Onde Aplicar:** Para adicionar responsabilidades/funcionalidades extras a um Usuario dinamicamente (ex: um UsuarioModerador).

- **Justificativa:** Permite adicionar novos comportamentos a objetos individualmente, de forma dinâmica, sem afetar o comportamento de outros objetos da mesma classe. Por exemplo, um usuário pode receber temporariamente permissões de moderador.

## 2. Adapter

- **Onde Aplicar:** Para integrar dados de uma API externa de astronomia.
- **Justificativa:** Converte a interface de uma classe em outra interface que os clientes esperam. Se os dados dos planetas vierem de uma API com um formato específico, um Adapter pode traduzir esses dados para a estrutura de classes do seu sistema (ex: sua classe Planeta).

## 3. Composite

- **Onde Aplicar:** Na representação de sistemas celestes (ex: um Planeta que contém uma lista de Luas).
- **Justificativa:** Compõe objetos em estruturas de árvore para representar hierarquias parte-todo. Permite que os clientes tratem objetos individuais e composições de objetos de maneira uniforme. Um planeta e suas luas podem ser tratados como um CorpoCeleste.

# Padrões Comportamentais (Behavioral)

Padrões que se concentram em algoritmos e na atribuição de responsabilidades entre objetos, facilitando a comunicação entre eles.

## 1. Observer

- **Onde Aplicar:** Para notificar usuários sobre novas atividades (ex: um novo Comentário em uma Postagem que o usuário segue).
- **Justificativa:** Define uma dependência um-para-muitos entre objetos, de modo que quando um objeto (o "observado" ou "subject") muda de estado, todos os seus dependentes (os "observadores") são notificados e atualizados automaticamente.

## 2. Strategy

- **Onde Aplicar:** Para permitir diferentes formas de ordenar o feed de Postagem (ex: por data, por curtidas, por relevância).
- **Justificativa:** Define uma família de algoritmos, encapsula cada um deles e os torna intercambiáveis. Permite que o algoritmo varie independentemente dos clientes que o utilizam.

## 3. Command

- **Onde Aplicar:** Para encapsular as ações de "curtir" e "não curtir" em Postagem e Comentário.
- **Justificativa:** Encapsula uma solicitação como um objeto, permitindo parametrizar clientes com diferentes solicitações, enfileirar ou registrar solicitações e suportar operações que podem ser desfeitas. As ações de curtir/descurtir podem ser objetos ComandoCurtir e ComandoDescurtir.

## 2. Proposta de Divisão em Grupos de Trabalho e Integração dos Padrões

A ideia é que cada um dos três grupos implemente um padrão de cada categoria (Criacional, Estrutural e Comportamental), focando em um módulo funcional coeso da aplicação.

### Grupo 1: O Núcleo do Fórum Social e Interações do Usuário

- **Foco do Módulo:** Construir a espinha dorsal da interação social no fórum: criação de conteúdo, definição de papéis de usuário e notificações sobre atividades.
- **Padrões a Implementar:**
  - Criacional: **Factory Method** (para criar Postagem e Comentário).
  - Estrutural: **Decorator** (para adicionar papéis/permisões a Usuario, como UsuarioModerador).
  - Comportamental: **Observer** (para notificar Usuarios sobre novos comentários em Postagens que seguem).
- **Justificativa da Integração:** Este grupo constrói um ciclo de interação social: o Factory Method cria o conteúdo; o Decorator define quem pode interagir e como; o Observer notifica sobre essas interações.

### Grupo 2: O Pipeline de Dados do Planetário e Visualização

- **Foco do Módulo:** Gerenciar a obtenção, adaptação e apresentação dos dados astronômicos que formam a base do planetário virtual.
- **Padrões a Implementar:**
  - Criacional: **Singleton** (para um ServicoPlanetario que gerencia o acesso aos dados dos planetas).
  - Estrutural: **Adapter** (para integrar dados de uma API externa de astronomia, adaptando-os ao modelo do sistema).
  - Comportamental: **Strategy** (para permitir diferentes formas de ordenar/filtrar a exibição dos planetas, ex: por tamanho, distância do sol).
- **Justificativa da Integração:** Este grupo foca no fluxo de dados: o Singleton centraliza o acesso; o Adapter busca e transforma os dados; a Strategy define como esses dados são processados para exibição.

### Grupo 3: Gerenciamento de Usuários, Estruturas Complexas e Ações Específicas

- **Foco do Módulo:** Lidar com a criação detalhada de usuários, a modelagem de estruturas hierárquicas do sistema (como sistemas planetários) e a execução de ações específicas do usuário.
- **Padrões a Implementar:**
  - Criacional: **Builder** (para a construção complexa de objetos Usuario).
  - Estrutural: **Composite** (para modelar o sistema solar como uma estrutura em

árvore: Planeta contendo Luas, ambos CorposCelestes).

- Comportamental: **Command** (para encapsular ações como "curtir" e "não curtir" Postagem ou Comentário como objetos).
- **Justificativa da Integração:** Este grupo foca na infraestrutura do usuário e suas interações: o Builder cria o usuário; o Composite modela os objetos complexos com os quais ele interage; o Command processa suas ações de forma desacoplada.

### Tabela Resumo da Distribuição

Grupo	Foco do Módulo	Padrão Criacional	Padrão Estrutural	Padrão Comportamental
1	Núcleo do Fórum Social e Interações	Factory Method	Decorator	Observer
2	Pipeline de Dados do Planetário e Visualização	Singleton	Adapter	Strategy
3	Gerenciamento de Usuários e Ações Específicas	Builder	Composite	Command