

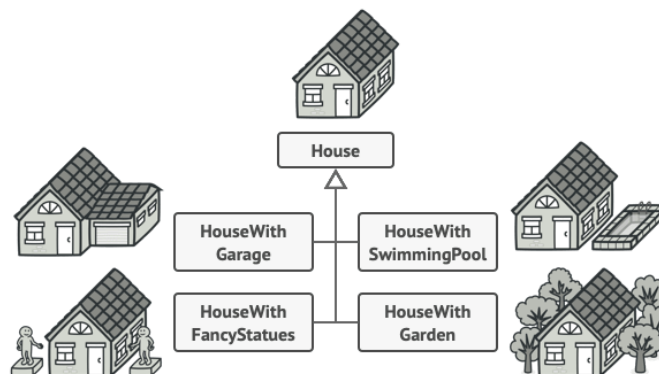
Padrões de Projeto Aplicados no Sistema de Gerenciamento de Pedidos de Cafeteria e Classificação de Corrida de Fórmula 1

1. Padrão Builder

O **Builder** é utilizado para simplificar a criação de objetos complexos, que possuem muitos atributos ou configurações, sem sobrecarregar o código com construtores longos ou múltiplos parâmetros.

O **Builder** foi escolhido porque:

- **Flexibilidade:** Permite construir objetos em etapas, garantindo que apenas os atributos necessários sejam preenchidos, sem forçar a definição de todos eles de uma vez.
- **Leitura de código mais clara:** Código mais legível, evitando construtores longos ou muitos parâmetros opcionais.
- **Facilidade de manutenção:** Adicionar ou remover atributos do objeto é mais fácil.

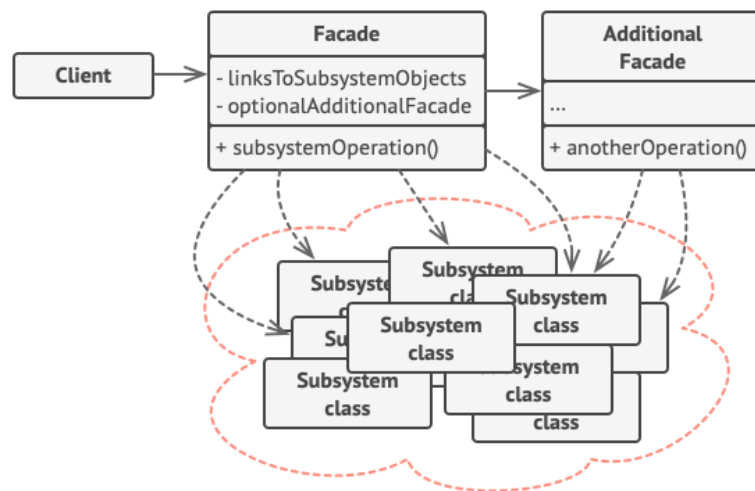


2. Padrão Facade

O **Facade** foi utilizado para **simplificar o acesso a subsistemas complexos** no **sistema de classificação de corrida**. Ele fornece uma interface única que abstrai a complexidade interna do sistema.

O **Facade** foi escolhido porque:

- **Simplicidade:** Facilita a utilização do sistema por outros módulos, fornecendo uma interface simples para operações complexas.
- **Baixo acoplamento:** Os componentes internos podem ser modificados sem afetar as partes externas, pois o **Facade** atua como intermediário.
- **Manutenção facilitada:** Qualquer alteração no funcionamento interno do sistema afeta apenas o **Facade**, sem impacto para quem utiliza a interface simplificada.



3. Padrão Singleton

O **Singleton** foi usado para garantir que **apenas uma instância de classes essenciais** seja criada em ambos os sistemas.

O **Singleton** foi escolhido porque:

- **Controle centralizado:** Garantir que haja uma única instância evita a criação de múltiplas conexões desnecessárias ou configurações conflitantes.
- **Consistência de dados:** Assegura que todas as partes do sistema estejam acessando a mesma instância de um recurso, evitando inconsistências.
- **Acesso global:** Fornece um ponto de acesso global para classes que devem ser únicas, facilitando a integração entre módulos.

