

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

FATEC DA ZONA LESTE

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Bruno Costa Rezende

Abstract Factory

São Paulo 2025

O Padrão Abstract Factory

O padrão Abstract Factory trata de criar famílias de objetos relacionados sem que o cliente precise conhecer suas implementações concretas. Em vez de criar os objetos diretamente, o sistema utiliza uma “fábrica” que sabe como montar cada conjunto de produtos, garantindo que todos sejam compatíveis entre si.

Princípio Fundamental

A ideia central é separar a lógica de criação dos objetos da sua utilização. Isso significa que, em vez de um módulo instanciar diretamente um objeto, ele solicita à fábrica a criação desse objeto. Assim, o cliente interage apenas com interfaces ou classes abstratas, sem depender das classes específicas que implementam a funcionalidade.

Como Funciona

1. **Interface da Fábrica:** Existe uma interface ou contrato que define os métodos de criação para cada tipo de objeto necessário. A fábrica é responsável por instanciar os produtos corretos de acordo com esse contrato.
2. **Fábricas Concretas:** Para cada família de objetos (por exemplo, para diferentes sistemas operacionais ou temas visuais), há uma fábrica concreta que implementa a interface. Cada uma dessas fábricas sabe exatamente como criar os objetos compatíveis com seu ambiente específico.
3. **Produtos Abstratos:** Os produtos são definidos por interfaces ou classes abstratas. Isso garante que o cliente interaja apenas com as definições gerais dos produtos e não com suas implementações específicas. Assim, o cliente pode trabalhar com qualquer variante de produto sem precisar alterar seu código.
4. **Cliente:** O cliente utiliza a fábrica para obter os produtos. Ele não se preocupa com os detalhes da criação, apenas com as funcionalidades que os objetos devem oferecer. Dessa forma, é possível trocar a fábrica (e, conseqüentemente, a família de produtos) sem modificar a lógica do cliente.

Benefícios do Padrão

1. **Desacoplamento:** Como o cliente não conhece as implementações concretas dos produtos, torna-se mais fácil modificar ou substituir partes do sistema sem afetar outras áreas.
2. **Consistência:** Garante que os produtos pertencentes a uma mesma família sejam compatíveis entre si, evitando conflitos e problemas de integração.
3. **Facilidade de Expansão:** Quando há necessidade de introduzir novos tipos de produtos ou variantes, basta criar novas fábricas concretas e suas respectivas implementações. O sistema se mantém flexível e facilmente extensível.

Aplicação na Prática

Imagine uma aplicação que precisa funcionar em diferentes sistemas operacionais. Em vez de condicionar a criação dos componentes a cada ambiente (como botões, menus, etc.), a aplicação utiliza uma fábrica que, dependendo do sistema detectado, cria os componentes corretos. Assim, toda a lógica de configuração e criação é centralizada na fábrica, enquanto a aplicação foca na utilização dos componentes.

Relação com os Princípios de Projeto

Esse padrão enfatiza a inversão de dependência, onde módulos de alto nível (como a interface de usuário) não dependem de módulos de baixo nível (a implementação dos componentes). Isso promove um design mais modular e flexível, onde as alterações em uma parte do sistema não necessariamente impactam outras partes.

Em resumo, o Abstract Factory oferece uma forma estruturada de criar famílias de objetos sem acoplamento direto entre a lógica de criação e a utilização dos objetos, facilitando a manutenção, escalabilidade e adaptação do sistema a diferentes contextos ou plataformas.