**Abstract Factory**

O Abstract Factory Method é um [padrão de Design Criativo](https://www.geeksforgeeks.org/design-patterns-set-1-introduction/) que permite produzir as famílias de objetos relacionados sem especificar suas classes concretas. Usando o Abstract Factory Method, temos as maneiras mais fáceis de produzir um tipo semelhante de muitos objetos.  
 Ele fornece uma maneira de encapsular um grupo de fábricas individuais. Basicamente, aqui tentamos abstrair a criação dos objetos dependendo da lógica, do negócio, da escolha da plataforma, etc.

## EstruturaPadrão de projeto Abstract Factory

1. **Produtos Abstratos** declaram interfaces para um conjunto de produtos distintos mas relacionados que fazem parte de uma família de produtos.
2. **Produtos Concretos** são várias implementações de produtos abstratos, agrupados por variantes. Cada produto abstrato (cadeira/sofá) deve ser implementado em todas as variantes dadas (Vitoriano/Moderno).
3. A interface **Fábrica Abstrata** declara um conjunto de métodos para criação de cada um dos produtos abstratos.
4. **Fábricas Concretas** implementam métodos de criação fábrica abstratos. Cada fábrica concreta corresponde a uma variante específica de produtos e cria apenas aquelas variantes de produto.
5. Embora fábricas concretas instanciam produtos concretos, assinaturas dos seus métodos de criação devem retornar produtos abstratos correspondentes. Dessa forma o código cliente que usa uma fábrica não fica ligada a variante específica do produto que ele pegou de uma fábrica. O **Cliente** pode trabalhar com qualquer variante de produto/fábrica concreto, desde que ele se comunique com seus objetos via interfaces abstratas.

## Aplicabilidade

## **Aplicabilidade**

Use o Abstract Factory quando seu código precisa trabalhar com diversas famílias de produtos relacionados, mas que você não quer depender de classes concretas daqueles produtos-eles podem ser desconhecidos de antemão ou você simplesmente quer permitir uma futura escalabilidade.

  O Abstract Factory fornece a você uma interface para a criação de objetos de cada classe das famílias de produtos. Desde que seu código crie objetos a partir dessa interface, você não precisará se preocupar em criar uma variante errada de um produto que não coincida com produtos já criados por sua aplicação.

* Considere implementar o Abstract Factory quando você tem uma classe com um conjunto de [**métodos fábrica**](https://refactoring.guru/pt-br/design-patterns/factory-method) que desfoquem sua responsabilidade principal.
* Em um programa bem desenvolvido cada classe é responsável por apenas uma coisa. Quando uma classe lida com múltiplos tipos de produto, pode valer a pena extrair seus métodos fábrica em uma classe fábrica solitária ou uma implementação plena do Abstract Factory.