O **Abstract Factory** é um padrão que permite que você produza famílias de objetos relacionados sem ter que especificar suas classes concretas.

Imagine que você esteja criando um simulador de loja de mobílias e o código consiste em classes que representam:

- Uma família de produtos relacionados, tais como: Cadeira, Sofa, MesaDeCentro.

- Várias variantes dessa família, por exemplo os produtos acima estão disponíveis nas variantes: Moderno, Vitoriano. ArtDeco.

Precisamos encontrar um modo de criar objetos de mobília individuais para que eles combinem com outros objetos da mesma família. Os clientes não gostam quando recebem mobília que não combina.

**Solução**: O padrão sugere que, antes de tudo, declaremos as interfaces para cada produto da família distinta de produtos (Cadeira, Sofa, MesaDeCentro). A partir disso, todas as variantes destes produtos devem seguir as interfaces criadas.

O próximo passo é criar uma fábrica abstrata, uma interface com um método de criação para cada um dos produtos de nossa família. Esses métodos retornam os tipos abstratos que definimos anteriormente: Cadeira, Sofa e MesaDeCentro.

Para finalizar, para cada variante de uma família de produtos, criamos uma nova fábrica, que irá herdar da fábrica abstrata. Assim teremos, por exemplo, uma FabricaFamiliaModerna, que irá retornar uma CadeiraModerna, SofaModerno ou MesaDeCentroModerna.

O código cliente tem que funcionar com ambos as fábricas e produtos a partir de suas interfaces abstratas. Isso permite a mudança do tipo de uma fábrica que passou para o código cliente, bem como a variante do produto que o código cliente recebeu, sem quebrar o código cliente atual.

**Prós:** Os prós são bem parecidos com o do FactoryMethod. Seguimos os princípios de *responsabilidade única* e *aberto/fechado*, além de evitarmos um vínculo forte entre os produtos concretos e o código cliente. Além disso, podemos ter certeza que os produtos que obtemos de uma fábrica são compatíveis entre si.

**Contras:** O código pode se tornar mais complicado do que deveria ser, uma vez que muitas novas interfaces são introduzidas junto com o padrão.