
Abstract factory

By Carlos Javier Bolaños Riascos



¿Qué? ¿Cuándo? ¿Por qué?

Cuando hacer uso del patrón FACTORY no es eficiente para controlar familias de objetos o productos, nace la propuesta de ABSTRACT FACTORY.

Es un patrón de diseño que permite hacer el control de familias de objetos grandes. Sirve para diferenciar entre familias y prevenir errores por mezcla de objetos sin relación (en este caso es la familia).

Producir familias de objetos sin especificar su clase concreta.

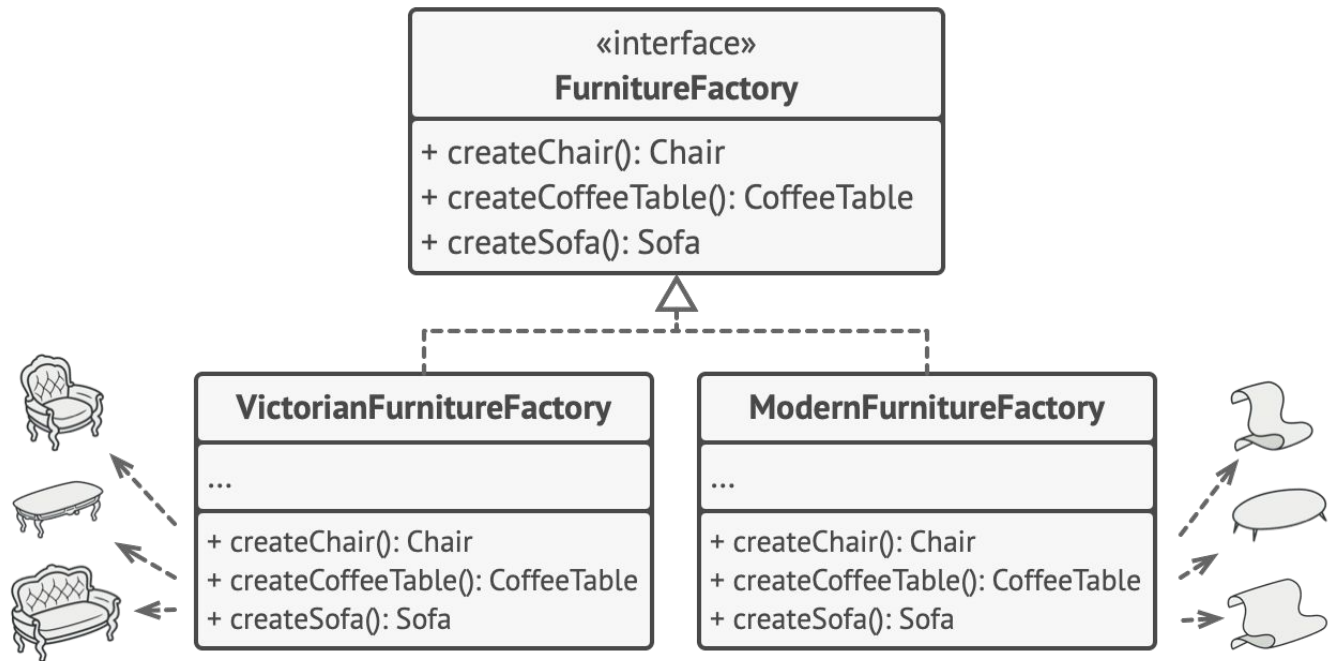
Vamos de lo general (abstracto) a lo concreto (particular).

Características...

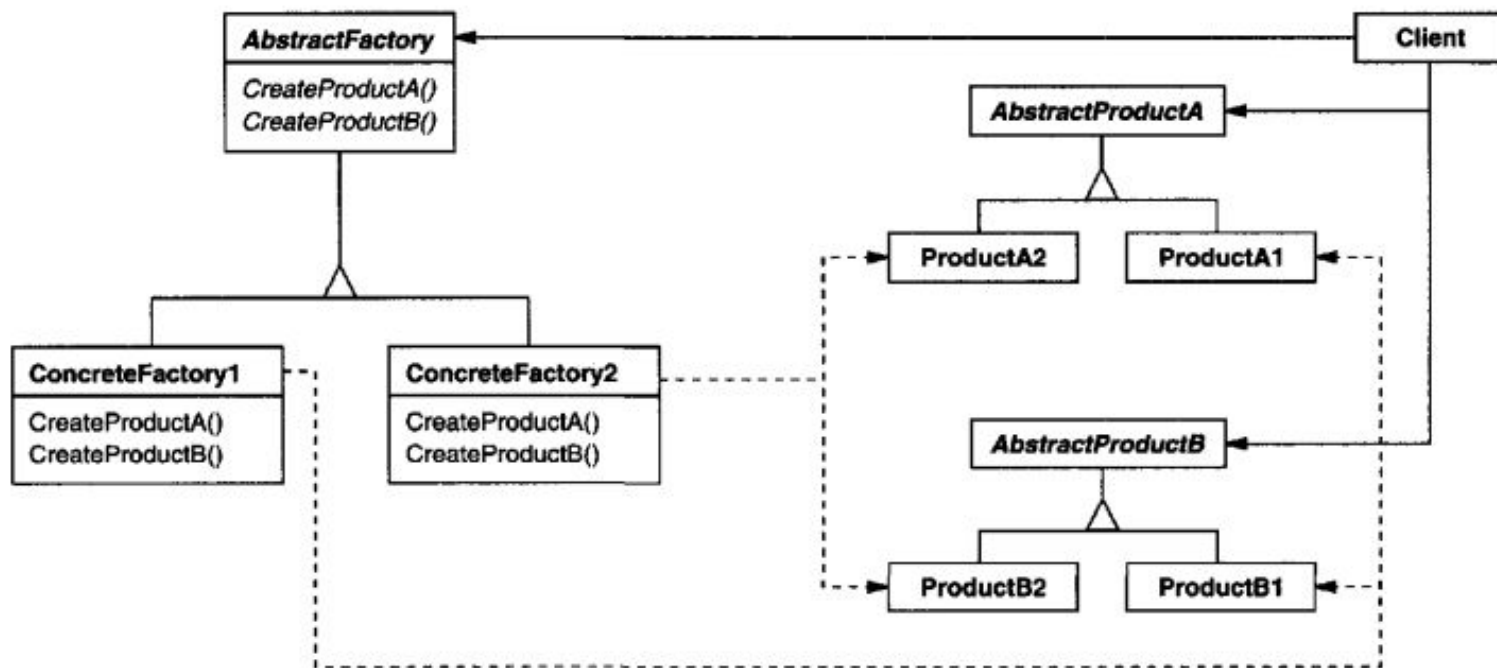
- Al ser un patrón de diseño creacional, se ve específicamente aplicado en el proceso de creación de objetos.
 - Se hace uso de interfaces, clases abstractas y polimorfismo.
 - Hay “roles” específicos para la creación de objetos.
 - Permite la diferenciación y organización de familias de objetos.
-

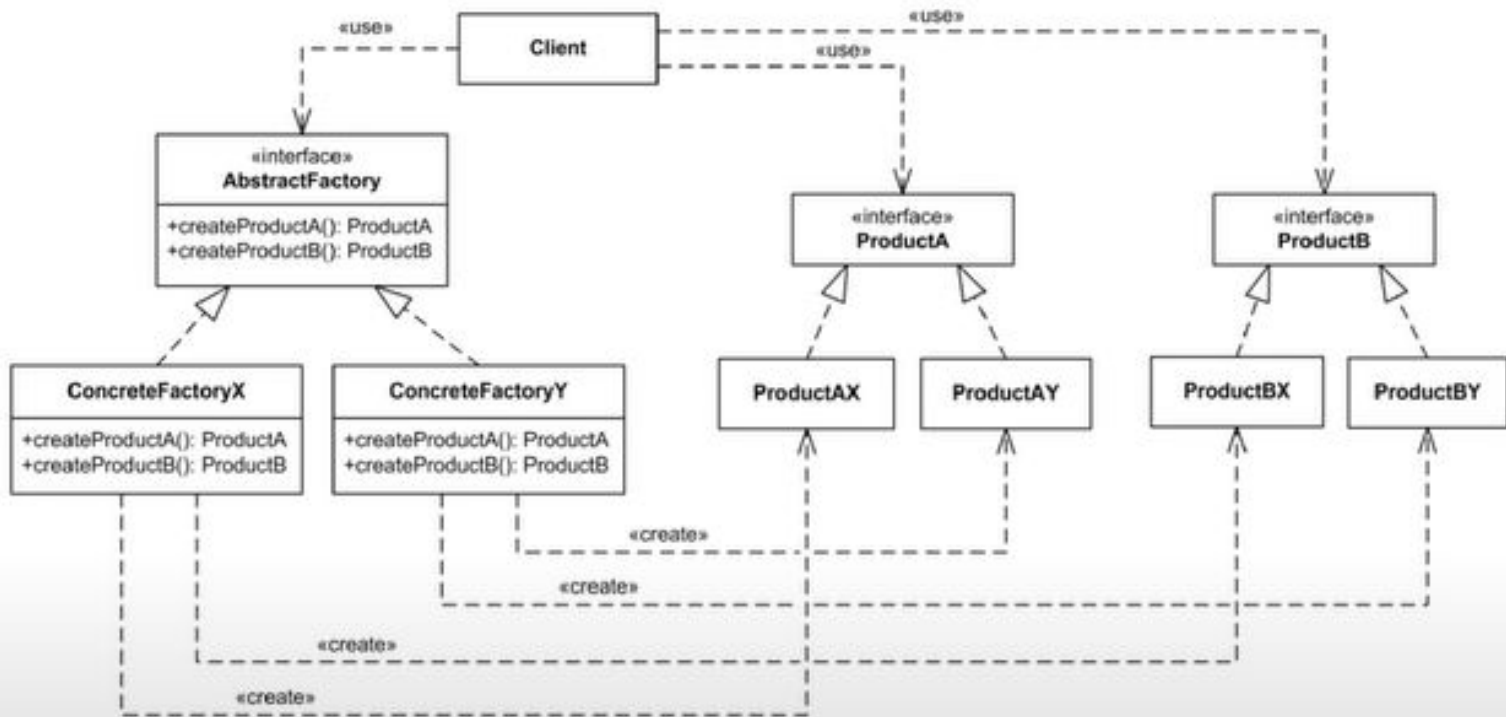
Participantes

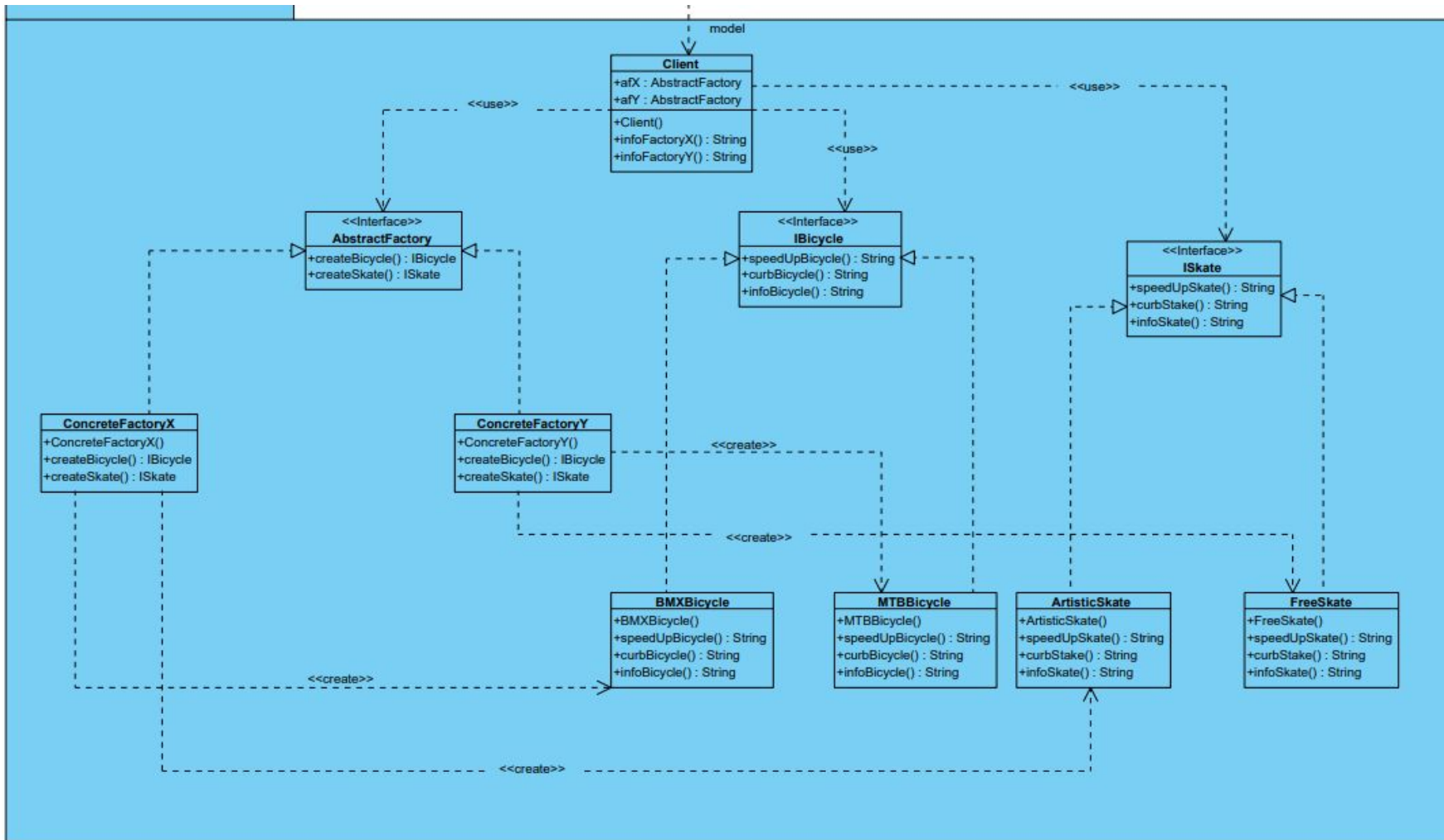
- **AbstractFactory:** Declara una interfaz de operaciones de creación de objetos de productos abstractos.
 - **ConcreteFactory:** Implementa las operaciones para crear objetos de productos.
 - **AbstractProduct:** Declara una interfaz para un tipo de objetos de productos.
 - **ConcreteProduct:** Define un objeto de producto creado por su correspondiente factoría. Implementa la interfaz Abstract product.
 - **Client:** Usa solo las interfaces declaradas por las clases AbstractFactory y Abstract product.
-



Structure







Mi implementación:

[Repositorio de Github](#)

Fuentes:

Factory:

<https://youtu.be/ILvYAzXO7Ek>

Abstract factory:

<https://youtu.be/CVlpiFJN17U>

- Betta Tech
-