

Patrones de Diseño

1. Factory (Fábrica)

Principios esenciales:

- Definir una interfaz para crear un objeto, pero permite que las clases derivadas decidan qué clase instanciar.
- Se utiliza para delegar la responsabilidad de instanciar objetos a una clase fábrica.

Desventaja:

- Puede llevar a una estructura de código complicada si hay demasiados tipos de productos o si la lógica de creación es compleja.

2. Abstract Factory (Fábrica Abstracta)

Principios esenciales:

- Proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar sus clases concretas.
- Las fábricas concretas que implementan esta interfaz producen objetos que siguen una variante de una familia de productos.

Desventaja:

- Si se necesita agregar nuevos tipos de productos a la fábrica, puede ser necesario modificar la interfaz de la fábrica abstracta, lo que afectará a todas las fábricas derivadas.

3. Builder (Constructor)

Principios esenciales:

- Separa la construcción de un objeto complejo de su representación para que el mismo proceso de construcción pueda crear diferentes representaciones.
- Permite producir diferentes tipos y representaciones de un objeto utilizando el mismo código de construcción.

Desventaja:

- Introduce más clases y complejidad en el código al separar el proceso de construcción de la representación del objeto.