

<<Code Slayers

Aguirre Morales Gael Alejandro 319165704  
Romero Cruz Fernando 319314256  
Sanchez Nava Rodrigo 319222571

Practica 04

//Factory//

-Principios:

Permite a las subclases elegir el tipo de objetos a crear.

El código solo se ocupa de las interfaces del producto; puede funcionar con cualquier clase de producto concreto definida por el usuario.

-Desventaja:

El código puede volverse más complejo ya que necesita introducir muchas subclases nuevas para implementar el patrón.

//Abstract Factory//

-Principios:

Se pueden introducir nuevas variantes de productos sin romper el código existente.

Ayuda a controlar las clases de objetos.

Evita el acoplamiento estrecho entre los productos concretos y el código del cliente

Los productos que obtiene de una fábrica son compatibles entre sí .

-Desventaja:

Extender fábricas abstractas por nuevos tipos de productos puede volverse complicado, ya que, fija el conjunto de productos que se pueden crear.

La compatibilidad con nuevos tipos de productos requiere agrandar la interfaz de fábrica.

//Builder//

-Principios:

Se puede construir objetos paso a paso, aplazar los pasos de construcción

Se puede introducir nuevas variantes de productos sin romper el código existente

Ayuda a minimizar la cantidad de parámetros en el constructor, dado eso, no es necesario pasar nulls para los parámetros opcionales al constructor

-Desventaja:

El numero de lineas de codigo aumenta al menos el doble.

Requiere crear un ConcreteBuilder separado para cada tipo diferente de producto.