

1. Menciona los principios de diseño esenciales del patrones Factory, Abstract Factory y Builder. Menciona una desventaja de cada patrón

Patrones de Creación

a) Factory



*Open-Closed.*

*" Objects or entities should be open for extension but closed for modification"*

extiende el código sin cambiarlo.



Una *desventaja* es que los clientes pueden heredar de la clase padre simplemente para crear un determinado objeto. La herencia esta bien cuando el cliente tiene que heredar de todos modos de la clase padre, pero si no es así estaría introduciendo una nueva vía de futuros cambios.

b) Abstract Factory



*Open-Closed.*

*" Objects or entities should be open for extension but closed for modification"*

**Abstract Factory** es la interfaz que provee un método para la obtención de cada objeto que pueda crear, de este modo si queremos crear estos objetos no implicaría estar modificando el código existente sino solo debemos extenderlo, en ningún momento se modifican las factorias concretas (familias de productos) cumpliendo así este principio.



Cuando se añaden nuevas familias de productos o cambian los existentes afecta a toda las familias creadas

c) Builder



Una *desventaja* se debe a que se crea más código siendo este menos legible y con ello aumenta la complejidad del programa, además requiere crear un ConcreteBuilder separado para cada tipo diferente de producto.

## Referencias

- [1] Gamma, E., Helm, R., Johnson, R. Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software.
- [2] [https://es.wikipedia.org/wiki/Abstract\\_Factory](https://es.wikipedia.org/wiki/Abstract_Factory)