**FACADE PATTERN**

**LOS PRINCIPIOS DE DISEÑO UTILIZADOS EN ESTE PATRÓN SON:**

* El principio abierto/Cerrado
* Principio de responsabilidad única
* Depende de abstracciones, no de implementaciones

**Los posibles casos de uso de este patrón serian.**

* Cuando tu código necesite trabajar con varias familias de productos relacionados, pero no deseas que dependa de las clases concretas de esos productos, pueden ser desconocidos de antemano o simplemente deseas permitir una futura extensibilidad.

**BENEFICIOS:**

* Puedes estar seguro de que los productos que se obtienen de una fábrica son compatibles entre sí.
* Encapsula la responsabilidad y el proceso de creación de objetos producto.
* El sistema estará adaptado para recibir futuras ampliaciones
* Reduce el acoplamiento de los productos con sus creadores.

**DESVENTAJAS:**

* El código puede volverse más complicado de lo que debería ser, ya que se introducen muchas nuevas interfaces y clases junto con el patrón.

**LAS PRINCIPALES DIFERENCIAS ENTRE EL ABSTRACTFACTORY Y EL FACTORYMETHOD SON:**

* El abstract hace uso de abstracciones (interfaces) para la implementación de sus fábricas mientras que el method hace uso de una extensión de un método para usarlo como fabrica.
* El abstract define familias de objetos mientras que el method define instancias únicas.
* El abstract tiene complicaciones al agregar sus nuevos objetos.
* El abstract tiene mas control sobre sus objetos creados.