

1. Menciona los principios de diseño esenciales de los patrones Decorator y Adapter. Menciona una desventaja de cada patrón.

Dado que ambos patrones pertenecen a la misma categoría (Patrones Estructurales) se cumplen los siguientes principios:

a) Decorator

- *Principio de responsabilidad única*  
Permite dividir funcionalidad entre clases con áreas de interés únicas.
- *Principio de abierto cerrado*
- Proporciona una manera más flexible de añadir responsabilidades a los objetos que la que podía obtenerse a través de la herencia(múltiple) estática, teniendo como una *desventaja* que se requiere crear una nueva clase para cada responsabilidad adicional, esto da lugar a muchas clases diferentes e incrementa la complejidad de un sistema. Heredar dos veces una clase resulta cuanto menos, propenso a errores

b) Adapter

- *Principio de responsabilidad única*  
Puede separar la interfaz o el código de conversión de datos de la lógica de negocio primaria del programa
- *Principio de abierto cerrado*  
Introduce nuevos tipos de adaptadores al programa sin descomponer el código cliente existente, siempre y cuando trabajen con los adaptadores a través de la interfaz con el cliente.
- Una *desventaja* es que la complejidad general del código aumenta, ya que se deben introducir un grupo de nuevas interfaces y clases. En ocasión resulta más sencillo cambiar la clase de servicios de modo que coincida con el resto del código.

## Referencias

- [1] Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software.