**Patrón observable:**

Este patrón permite monitorear el valor/estado de un objeto o variable, para que al ser cambiado se les informe a los demás objetos, que están suscritos, que se ha complementado la ejecución de un evento sobre el cual se modificó el valor y puedan realizar alguna acción. Es muy útil en MVC para que el backend esté pendiente del cambio sobre una vista.

**Patrón decorador:**

Este patrón permite añadir funcionalidades a objetos o clases sin modificar su estructura, es decir sin modificar su definición interna genérica, agregando anotaciones que internamente implementan la funcionalidad deseada.

**Patrón singleton:**

Cuando se define una clase cuya estructura hará que sus instancias sean completamente idénticas y que se necesita en varias partes del código, el patrón singleton ayuda a encapsular las funcionalidades en único objeto que podrá ser usado de manera global en cualquier parte del código sin crear instancias iguales. En este sentido puede ser beneficioso cuando se desea acoplar la conexión entre varias capas del software.