Resumen principios de diseño

SOLID

**S(**Single Responsibility Principle**):** Este principio se basa en una clase solo debe tener una única razón para cambiar, así que se debe reunir las cosas que cambian por la misma razón y separar las que la hacen por una diferente.

**O(**Open/Closed Principle**):** Se dice de este principio que las clases deberían de ser capaz de estar abiertas para extenderse, mas no deben ser modificadas, por lo que son cerradas en este aspecto (de ahí el cerrado/abierto del nombre del principio).

**L(**Liskov Substitution Principle**):** La base de este principio dicta que las clases derivadas de un programa podrían ser sustituidas por su clase base, esto comprobando que la funcionalidad del programa no se ve comprometida.

**I(**Interface Segregation Principle**):** Hay que hacer determinadas interfaces para una finalidad concreta, ya que es preferible contar con numerosas interfaces para métodos específicos que pocas interfaces para muchos métodos.

**D**( Dependency Inversion Principle **):**