SOLID

Principio de Responsabilidad Única (SRP - Single Responsibility Principio) : Cada clase o módulo debe tener una única razón para cambiar. Esto significa que una clase debe tener una sola responsabilidad y no debería haber más de una razón para modificarla.

Principio de Abierto/Cerrado (OCP - Open/Closed Principio) : Las entidades de software (clases, módulos, etc.) deben estar abiertas para extensión pero cerradas para modificación. Esto significa que se deben poder agregar nuevas funcionalidades sin cambiar el código existente.

Principio de Sustitución de Liskov (LSP - Liskov Substitution Principio) : Las instancias de una clase derivada deben poder ser sustituidas por instancias de la clase base sin afectar la integridad del programa. En otras palabras, las subclases deben ser intercambiables por sus clases base sin causar problemas.

Principio de Segregación de Interfaz (ISP - Interface Segregation Principio) : Los clientes no deben depender de interfaces que no utilizan. Es mejor tener interfaces específicas y especializadas para cada cliente en lugar de una única interfaz grande que abre muchas funcionalidades.

Principio de Inversión de Dependencia (DIP - Principio de Inversión de Dependencia) : Los módulos de alto nivel no deben depender de módulos de bajo nivel. Ambos deben depender de abstracciones. Además, los detalles no deben depender de abstracciones, sino al revés.