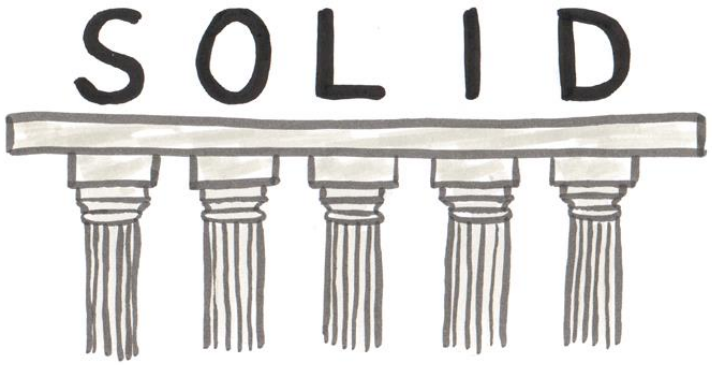
Principios SOLID

Definición

Serie de buenas practicas que guían la forma de diseñar un sistema, no es 100% requerido en todos los programas, se pueden usar unos y otros no, ser más flexible o menos dependiendo de lo que el programa requiera.

- Single Resposibility.
- Open Closed.
- Liskov Substitution.
- Interface segregation.
- Dependency Inversion.



OBJETIVOS

- Mas mantenible.
- Fácil aplicar cambios.
- Más legible y fácil de entender.
- Menos propenso a errores y más fácil arreglarlos.

Single Resposiblity (principio de responsabilidad única) SRP

Indica que cada clase o metodo debería tener una(1) sola responsabilidad, debería encargarse de una sola parte del sistema, de esta forma se asegura que cada clase realiza su tarea de manera efectiva.
Si solamente tiene un objetivo este será más facil de cumplir.

Open Closed (Principio de abierto cerrado) OCP

El codigo debe estar abierto para añadirle nuevas funciones pero cerrado para su modificación.
Esto hace que el sistema esté pretegido y sea más dificil de romper. Para añadir nuevas funciones se tiene que escribir codigo nuevo.
La unica razón por la cual se debería tocar el codigo existente se para arreglar errores, de manera contraria no se debería tocar.

Para lograr esto las herramientas más obvias son la HERENCIA y el POLIMORFISMO.

Liskov substitution (Principio de sustitución de Liskov) LSP

Una clase que es hija de otra clase debe poder usarse como si fuese el padre sin alterar su el correcto funcionamiento del desarrollo.

Básicamente que el hijo que hereda de una clase tenga que ser capaz de utilizar todos sus métodos y propiedades, en caso contrario lo mejor sería crear diferentes interfaces y que las 2 clases implementen las que necesiten.

Interface segregation (Principio de segregación de la interfaz) ISP

Es mejor tener muchas interfaces con metodos especificos que una interfaz o clase con una gran cantidad de metodos que tal vez no vaya a usar.

Las clases implementarán tantas interfaces como necesiten, pero no tendrán metodos de incio que tal vez puedan no necesitar

Dependency inversion (Principio de inversión de la dependencia) DIP

Los módulos de alto nivel no deben depender de los módulos de bajo nivel, ambos deberían depender de abstracciones.

Alto nivel se refiere a operaciones cuya naturaleza es más amplia o abarca un contexto más general y bajo nivel son componentes individuales más específicos.