

Principios SOLID



● Principios SOLID

- **Directrices** para **diseñar** software orientado a objetos
- Facilitan la creación de sistemas más **robustos, flexibles y mantenibles**
- Reducción de **complejidad**, mejora en la **mantenibilidad y escalabilidad** del código
- Diseño más **limpio y modular**, facilitando la evolución y adaptación del software
- Promueven un enfoque de diseño que **reduce el acoplamiento y mejora la cohesión**
- Permiten una mayor facilidad para agregar **nuevas funcionalidades** y corregir errores **sin afectar el sistema existente**

- Principios SOLID - **SRP**

- SOLID

- **Single Responsibility Principle (SRP)**
 - Una sola responsabilidad



SINGLE RESPONSIBILITY PRINCIPLE

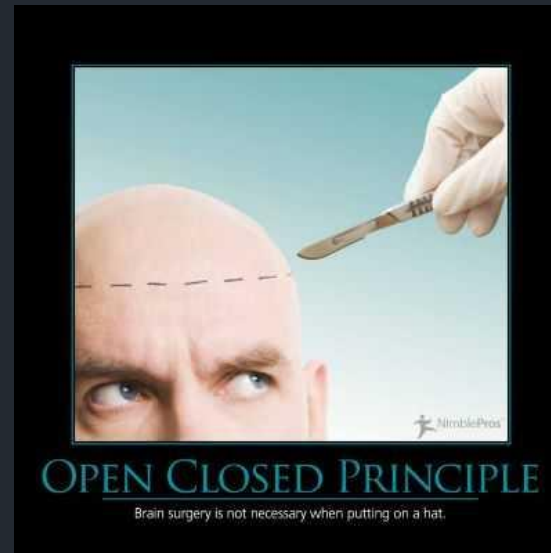
Just Because You Can, Doesn't Mean You Should

- Principios de Diseño Software – SOLID - OCP

- SOLID

- Open/Close Principle (OCP)

- Abierto para su extensión, pero cerrado para su modificación

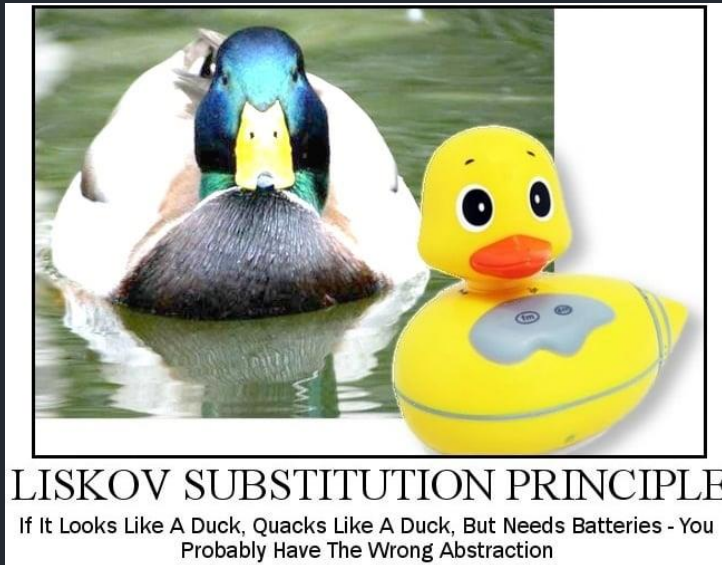


- Principios de Diseño Software – SOLID - LSP

- SOLID

- Liskov Substitution Principle (LSP)

- Una clase derivada no debe modificar el comportamiento de la clase base



Principios de Diseño Software – SOLID - ISP

◦ SOLID

▫ Interface Segregation Principle (ISP)

- Una clase que implementa una interfaz no debe depender de métodos que no utiliza.



INTERFACE SEGREGATION PRINCIPLE

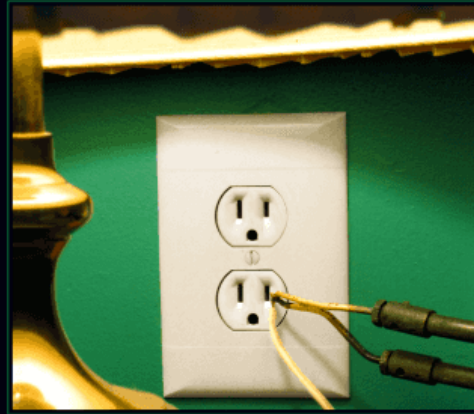
Don't force the client to depend on things they don't use.

- Principios de Diseño Software – SOLID - **DIP**

- SOLID

- **Dependency Inversion Principle (DIP)**

- Las clases de alto nivel no deben depender de clases de bajo nivel



DEPENDENCY INVERSION

Would you solder a lamp directly to the electrical wiring in a wall?