**Principios SOLID**

Los principios **SOLID** son un conjunto de buenas prácticas para el diseño de software orientado a objetos. Su objetivo principal es hacer que el código sea más fácil de entender, mantener y extender, evitando dependencias rígidas y diseños frágiles.

**Principios**

**S - El Principio de responsabilidad única (Single Responsibility Principle)**Cada clase debe tener una sola razón para cambiar, es decir, una única responsabilidad o propósito.

**O - El Principio Abierto-Cerrado (Open-Closed Principle)**  
El software debe estar abierto para la extensión, pero cerrado para la modificación. Se pueden agregar nuevas funcionalidades sin cambiar el código existente.

**L - El Principio de sustitución de Liskov (Liskov Substitution Principle)**  
Las clases derivadas deben poder sustituir a sus clases base sin alterar el comportamiento esperado del programa.

**I - El Principio de segregación de interfaz (Interface Segregation Principle)**  
Es mejor tener muchas interfaces específicas y pequeñas que una sola interfaz grande y general. Así, las clases implementan solo lo que realmente necesitan.

**D - El Principio de inversión de dependencia (Dependency Inversion Principle)**  
Los módulos de alto nivel no deben depender de módulos de bajo nivel, ambos deben depender de abstracciones (interfaces). Además, las abstracciones no deben depender de detalles, sino que los detalles deben depender de abstracciones.