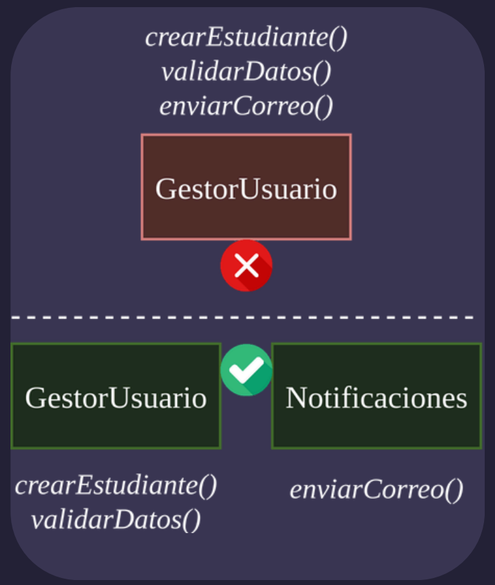
Principios SOLID

Principios para crear software robusto y adaptable, con arquitectura eficiente y código bien organizado en clases o estructuras de datos.

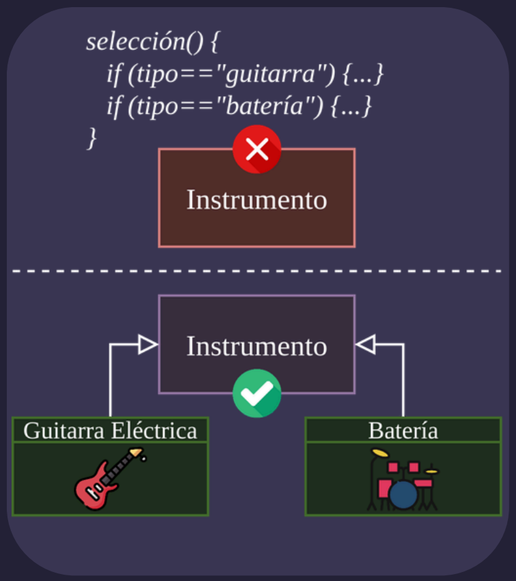
S
Single Responsability

«Un módulo debe encargarse de una sola función o grupo específico de tareas, por lo que debe tener una única razón para cambiar»



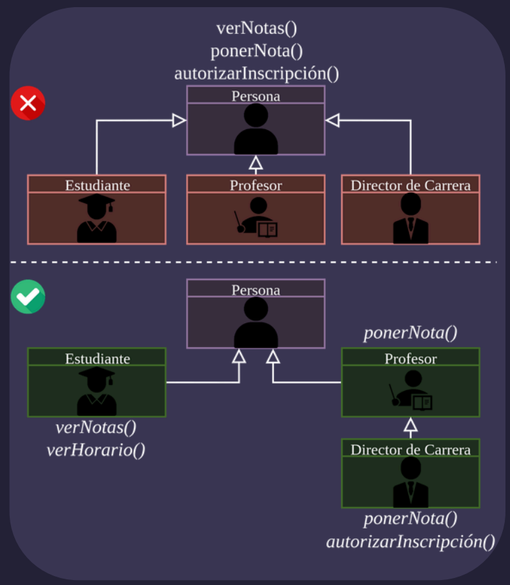
O
Open/Closed

«El código debe estar abierto para su extensión, pero cerrado para su modificación»



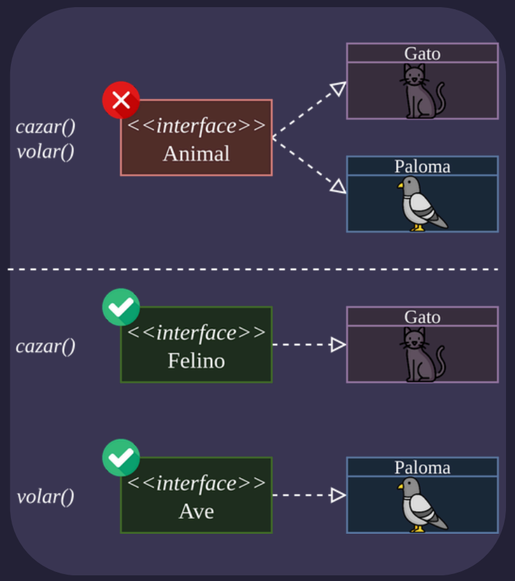
L
Liskov Substitution

«Las subclases deben poder reemplazar a sus clases base sin romper el programa»



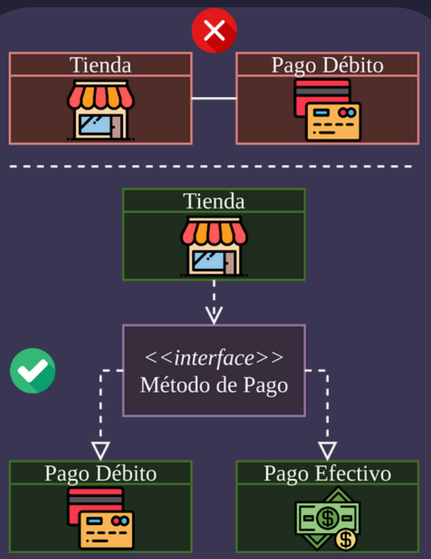
I
Interface Segregation

«Las interfaces deben dividirse en partes pequeñas y específicas, evitando métodos innecesarios para ciertas clases»



D
Dependency Inversion

«El código depende de abstracciones y no de implementaciones concretas»



Inspirado en X Julián Campos (2025)