



# Universidad de Costa Rica

## sede de occidente

# S O L I D

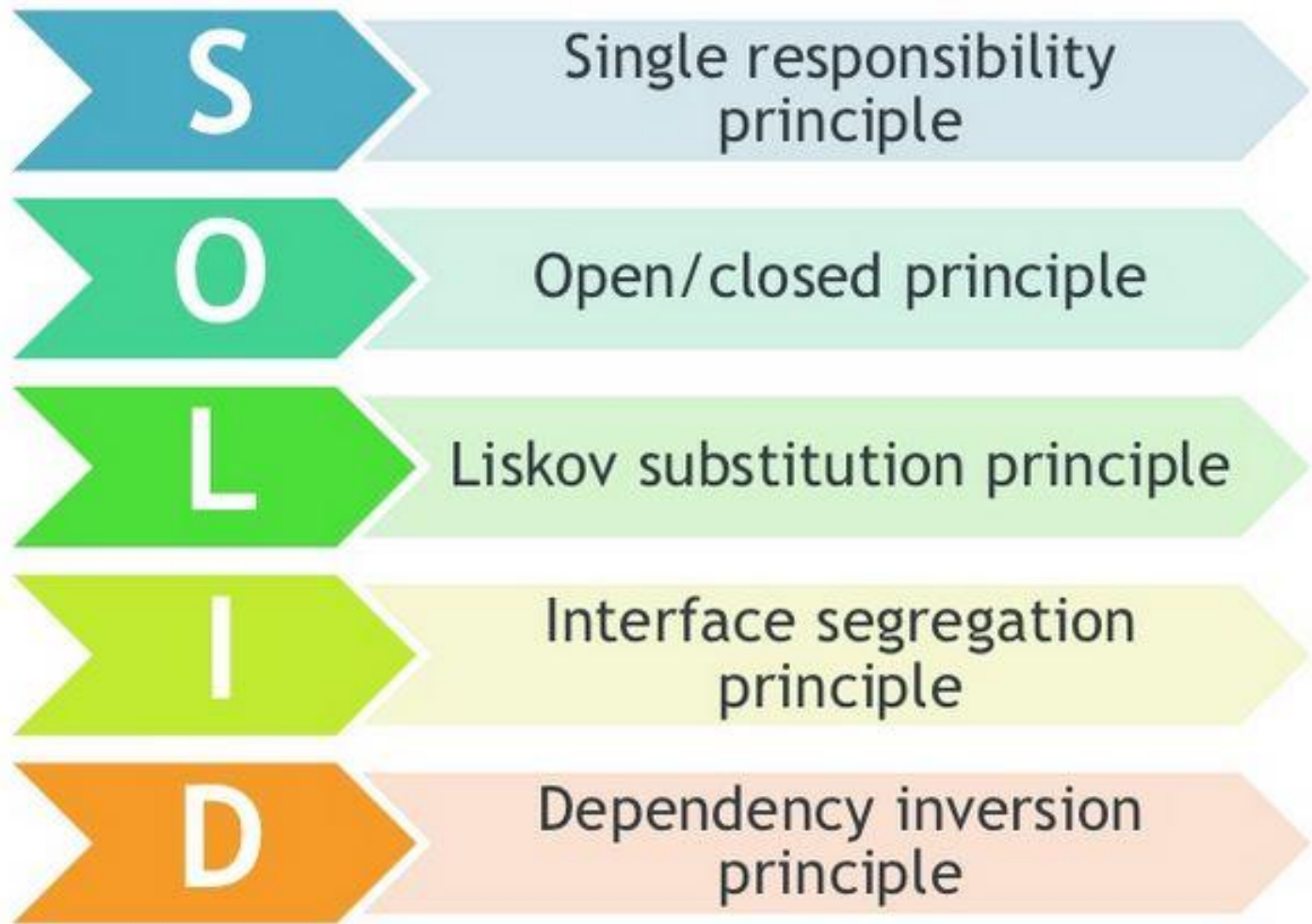
p r i n c i p l e s



# SOLID

Software Development is not a Jenga game

# ¿Qué es?



# SRP | Single responsibility principle

- Una clase debe tener solo una razón para cambiar (Martin. R.).
- Facilidad de Cambio.
- Funcionalidad.
- Cohesión alta. /acoplamiento bajo.

# OCP | Open/close principle

- Abierta a extensión, cerrado a modificación.
- Modularidad.
- El comportamiento debe poder ser alterado sin tener que modificar el código.

# LSP | Liskov substitution principle

- Si una función recibe un parámetro de tipo  $X$ , pero le enviamos otro de tipo  $Y$ , que hereda de  $X$ , dicha función debe proceder correctamente.

# ISP | Interface segregation principle

- El cliente no debería estar obligado a depender de interfaces que no utilizan.
- La interfaz no debe estar cargada de métodos.

# DIP | Dependency inversion principle

- Módulos de alto nivel no deben depender de módulos de bajo nivel.
- Abstracciones no deben depender de detalles, si no viceversa.



# Entonces, ¿Qué es S.O.L.I.D.?

Es un acrónimo de:

- **S**iempre
- **O**lvido
- **L**o
- **I**nteressante del
- **D**esarrollo