



IIC2113 - Diseño Detallado de
Software

Principios Fundamentales de la Programación

Diego Passi

M. Fernanda Sepúlveda

Jaime Yañez

Índice

1. Motivación
2. ¿Qué son los Principios Fundamentales?
3. Curly's Law
4. Command Query Separation
5. Boy-Scout Rule
6. KISS
7. DRY
8. YAGNI

1.

Motivación

Motivación

Deberíamos tener ciertas ideas o prácticas internalizadas.



2.

¿Qué son los
principios
fundamentales?

¿Qué son los principios fundamentales?

Son prácticas que ayudan a desarrollar programas **robustos**, **mantenibles** y que se puedan **modificar**.

- ▣ Abstracción
- ▣ Ocultamiento
- ▣ Cohesión
- ▣ Acoplamiento

▣ SOLID

- ▣ Simple Responsibility
- ▣ Open-Closed
- ▣ Liskov Substitution
- ▣ Interface Segregation
- ▣ Dependency Inversion

3.

Curly's Law

Curly's Law: Do One Thing

Una variable debe tener **un y sólo un significado**, este debe ser el mismo durante todo el código.

Curly: Do you know what the secret of life is?

Curly: This. [holds up one finger]

Mitch: Your finger?

Curly: One thing. Just one thing. You stick to that and the rest don't mean shit.

Mitch: But what is the "one thing?"

Curly: [smiles] That's what *you* have to find out.



4.

Command Query Separation

Command Query Separation

Cada método debe ser un comando que ejecute una acción en específico o una consulta que retorne información, pero no ambas al mismo tiempo.

Command: Cambia el estado observado de un sistema pero no retorna un valor.

Query: Retorna un resultado y no cambia el estado observado de un sistema.

5.

Boy-Scout Rule

Boy-Scout Rule

Al hacer cambios a código existente se tiende a degradar la calidad del mismo debido a la acumulación de falencias técnicas.

Estas falencias son suplidas por un Refactoring Continuo

“Leave the code cleaner than we found it”



6.

KISS

KISS: Keep It Simple Stupid!

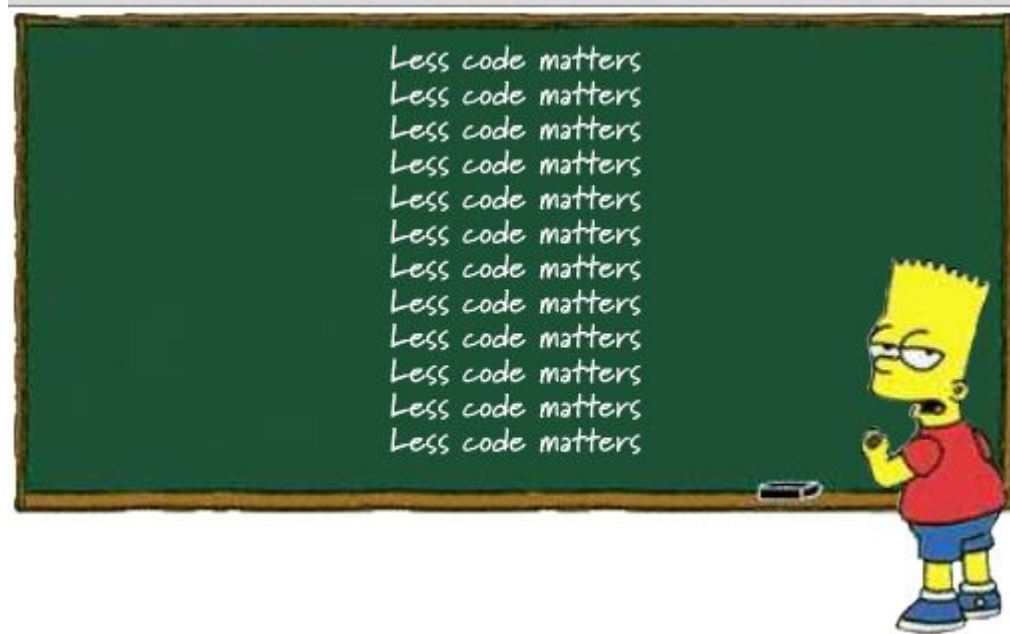
El evitar la complejidad y buscar la simplicidad debe ser siempre una meta fija.

El código simple toma menos tiempo para escribirse, tiene menos bugs y es más fácil de modificar.

7.

DRY

DRY: Don't repeat yourself



DRY: Don't repeat yourself

Cada pieza dentro de un sistema debe tener una única e inequívoca representación dentro de un sistema.

- Duplication is waste
- Repetition in process calls for automation
- Repetition in logic calls for abstraction

Regla de Tres:

El código puede ser copiado. Pero cuando se tiene el mismo código copiado 3 veces: entonces debe ser extraído a un nuevo procedimiento.

8.

YAGNI

YAGNI: You Ain't Gonna Need It



YAGNI

It may look like overkill, but I'm sure we'll need it eventually.



YOU AIN'T GONNA NEED IT

Don't waste resources on what you *might* need.

YAGNI: You Ain't Gonna Need It

No implementes algo a menos que sea necesario.

- Cualquier trabajo que signifique una feature del futuro, implica perder esfuerzo en para las necesidades de la iteración actual.
- Parece que el código se hincha: se convierte cada vez en algo más grande y complicado.

Zero One or Infinity Rule

No deben ser permitidos límites arbitrarios para el número de instancias de cualquier entidad.

- Una entidad no debe ser permitida, debe existir una única, o debe permitirse que exista cualquier número de ellas.
- La lógica del programa debe restringir situaciones borde. No debe forzar un “hard limit”.

Referencias



Own production examples:

- ▣ <https://github.com/mf222/principles-code>



wikipedia.org

- ▣ https://en.wikipedia.org/wiki/Zero_one_infinity_rule



programmer.97things.oreilly.com

- ▣ http://programmer.97things.oreilly.com/wiki/index.php/Don't_Repeat_Yourself



martinflower.com

- ▣ <http://martinfowler.com/bliki/CommandQuerySeparation.html>



blog.codinghorror.com

- ▣ <https://blog.codinghorror.com/curlys-law-do-one-thing/>

¿Preguntas?

iGracias por su atención!