

Apuntes de ingeniería del software

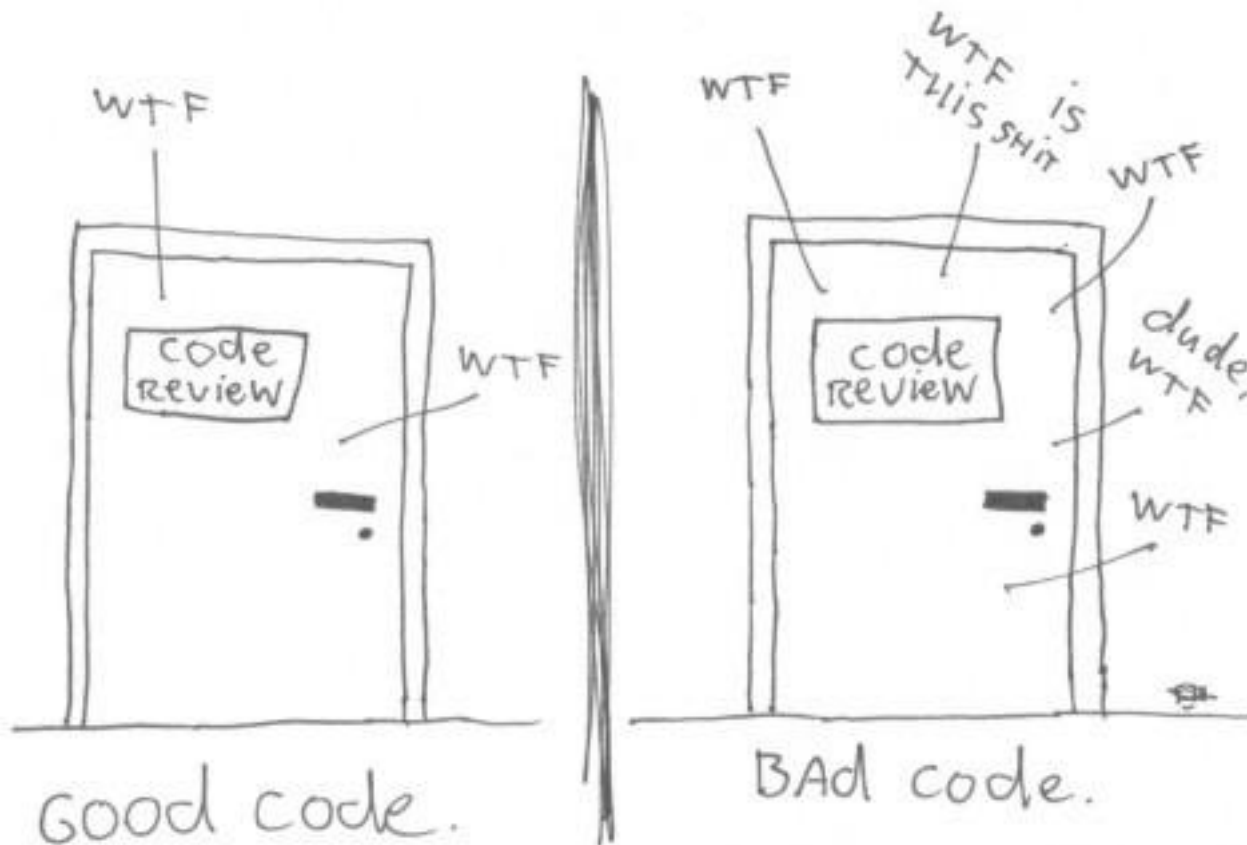
# Cualidades del código limpio

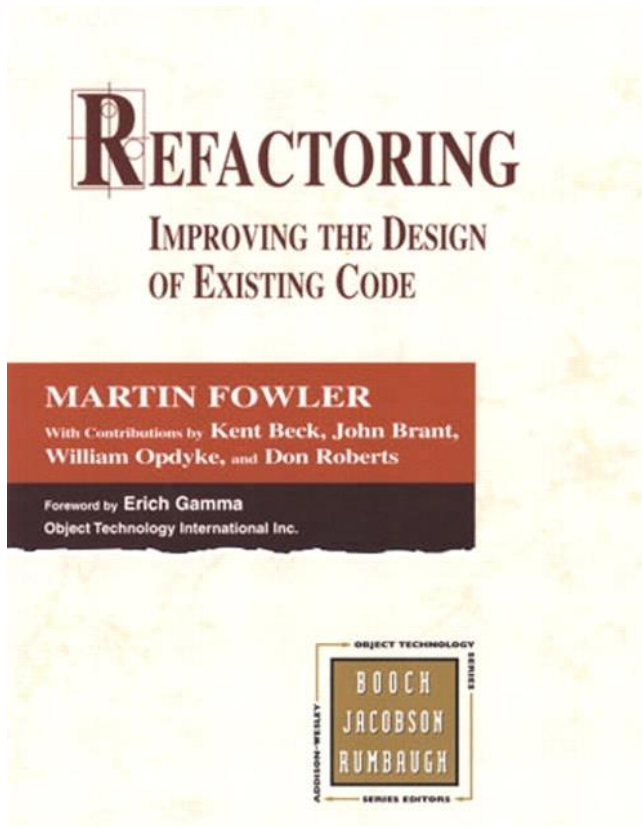
---

José Juan Hernández Cabrera

Escuela Universitaria de Ingeniería Informática. Universidad de  
Las Palmas de Gran Canaria  
Enero 2014

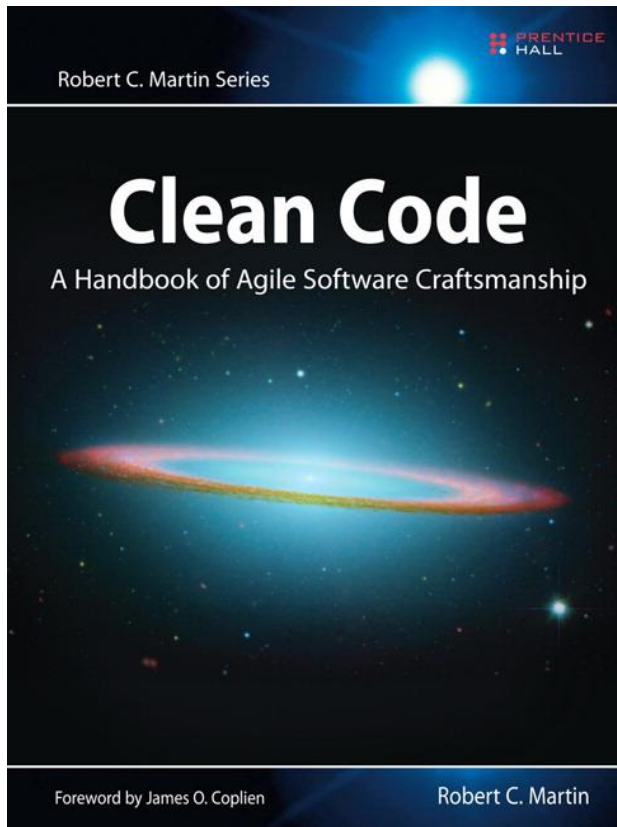
The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE





Este libro nos ayuda a desarrollar el olfato y detectar “código apestoso” (code smell)

Refactorizar debe de ser una tarea cotidiana



Hay dos motivos por los que leer este libro:

El primero, **eres programador**

El segundo, **quieres ser mejor programador**

# Cualidad 1

**El código limpio no hace demasiadas cosas, el código limpio es enfocado.**

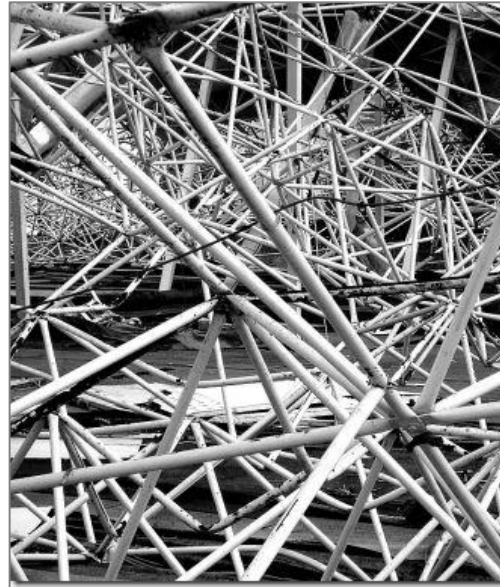
Cada clase, cada método o cualquier otro tipo de modulo debería cumplir con el principio de responsabilidad única. Un módulo sólo debiera ser responsable de un único aspecto de los requisitos del sistema



# Cualidad 2

**El código limpio no usa rodeos ni soluciones ofuscadas**

La lógica debe ser directa, clara, para que a los fallos les sea difícil esconderse.



# Cualidad 3

## El código limpio no es redundante

Cumplir con la regla DRY  
(Don't Repeat Yourself, o No  
Te Repitas).

Si se aplica correctamente  
DRY, los cambios de los  
requisitos solo debieran  
obligar a realizar cambios en  
un único lugar.



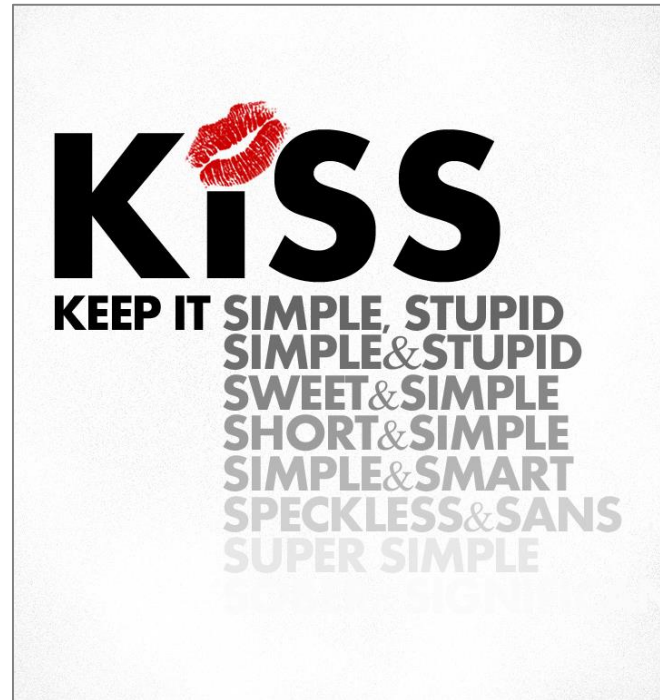


# Cualidad 4

## El código limpio es placentero de leer

Para lograr esto debemos cumplir con el principio KISS (Keep It Simple, Stupid!)

Se debe evitar la complejidad innecesaria. Esto es una práctica que nos alienta a enfocarnos exclusivamente en las cosas más simples que hagan funcionar al software.





# Cualidad 5

**El código limpio puede ser modificado fácilmente por cualquier otro desarrollador**

No escribimos código para nosotros mismos, o para el compilador. Escribimos código para otro desarrollador.

Además, en pocos meses nosotros mismos somos "otro desarrollador".



# Cualidad 6

## El código limpio debe tener dependencias mínimas

Mientras más dependencias tenga, más difícil va a ser de mantener y cambiar en el futuro.



# Cualidad 7

## El código limpio es pequeño

Tanto las clases como los métodos deberían ser cortos, preferentemente con pocas líneas de código. Debe estar bien dividido. Mientras más dividamos el código, más fácil se vuelve digerirlo.



# Cualidad 8

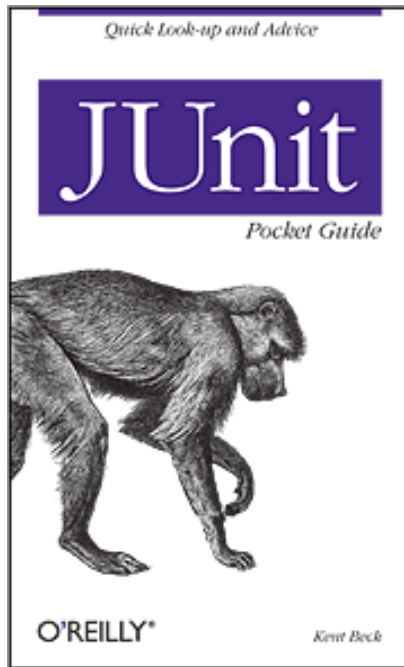
## El código limpio es expresivo

Los nombres de los módulos  
deben ser significativos.  
No tienen que resultar  
engañosos.  
Tienen que ser distintivos.  
El código se debe documentar  
a si mismo.



# Cualidad 9

El código limpio tiene módulos de prueba



**JUnit**