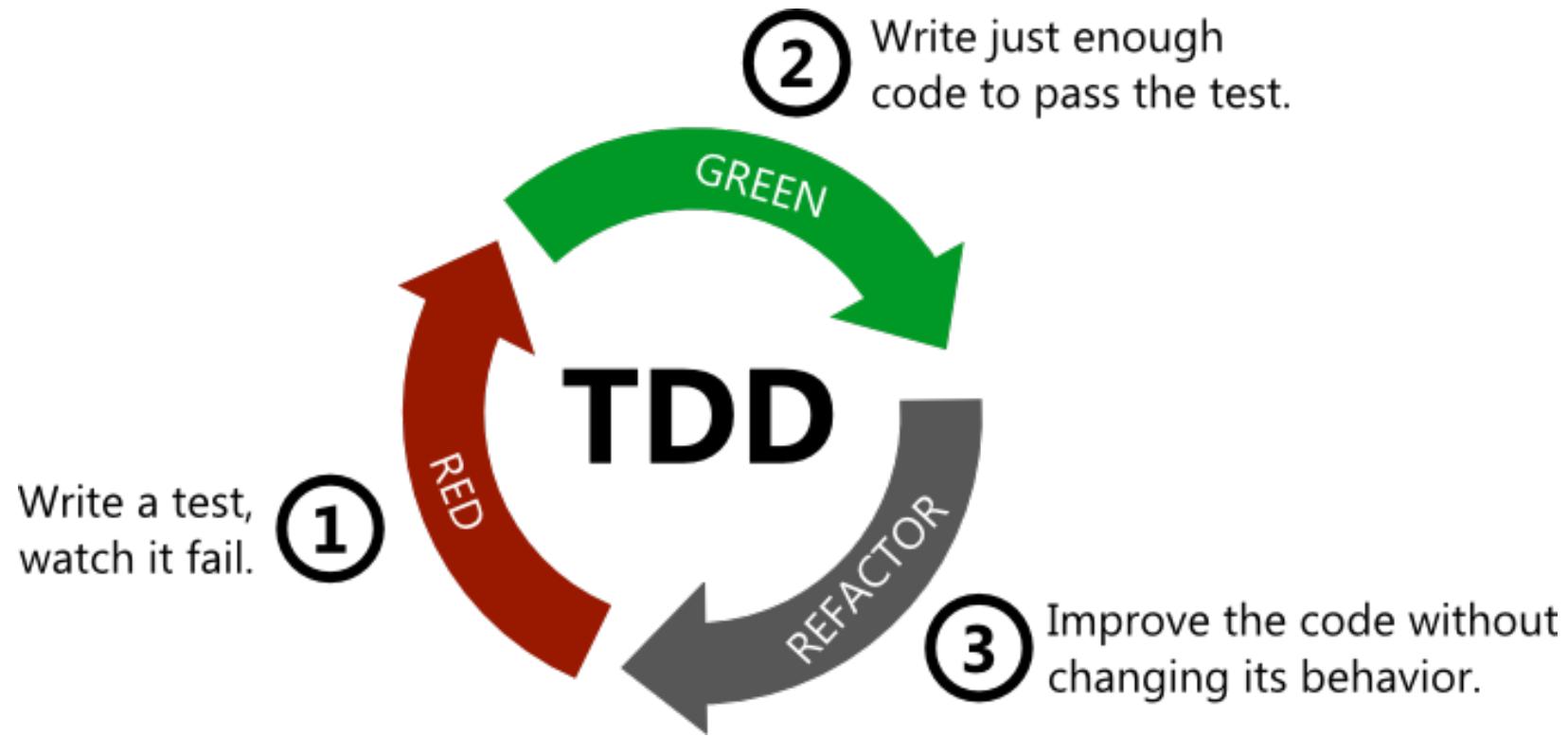
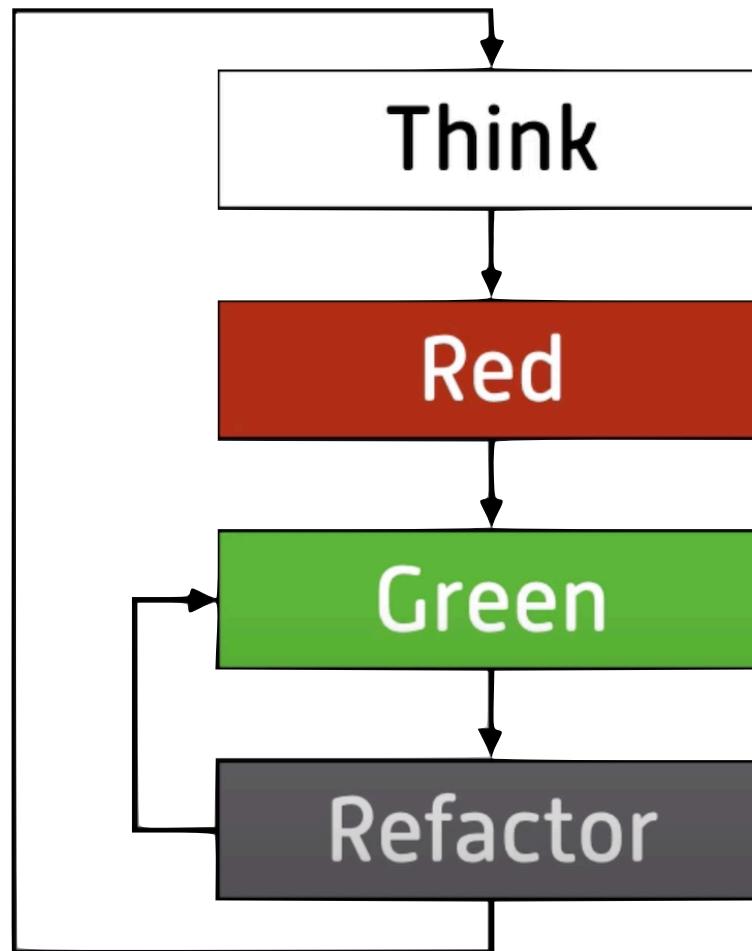


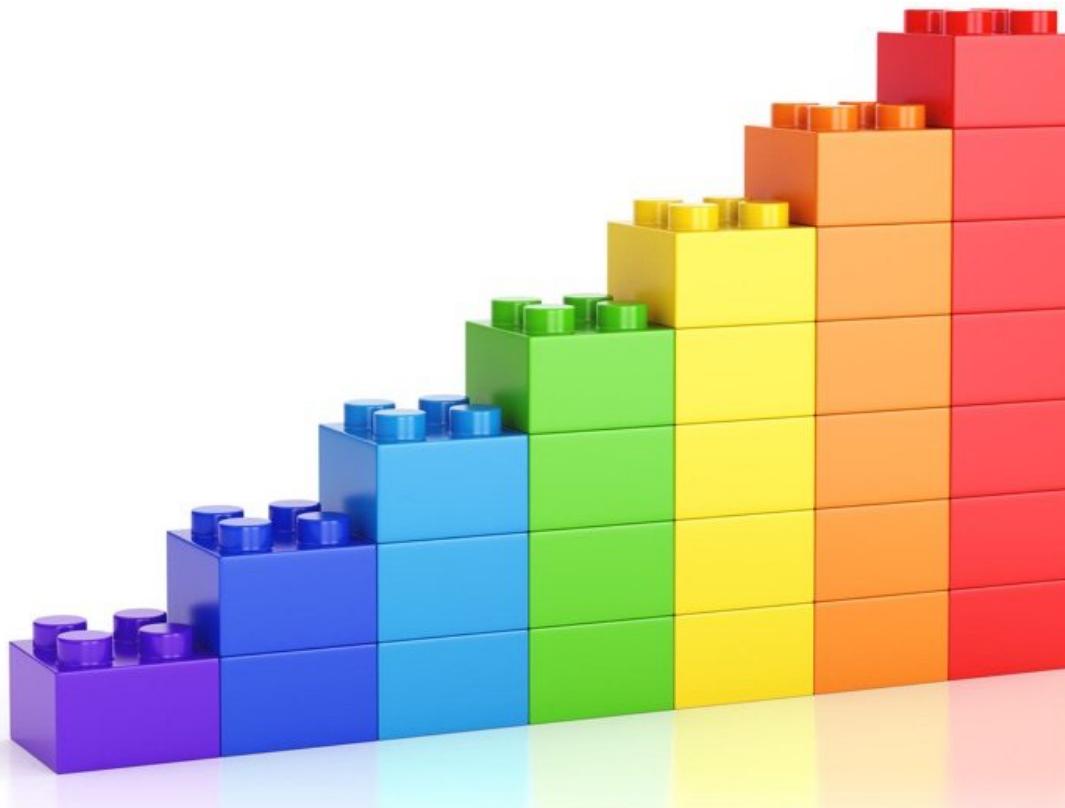
**MADS  
S7: TDD  
(Bloque 2 - XP)**



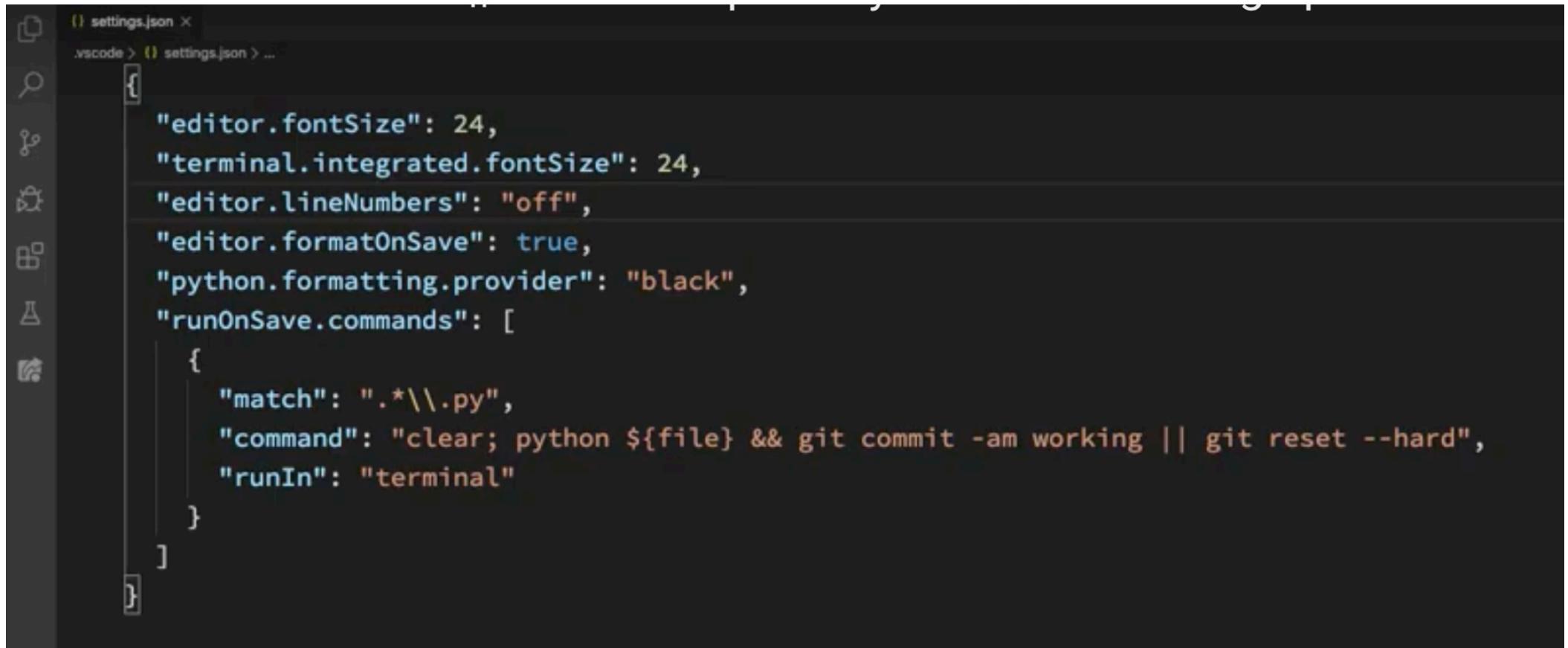




# Pequeños incrementos

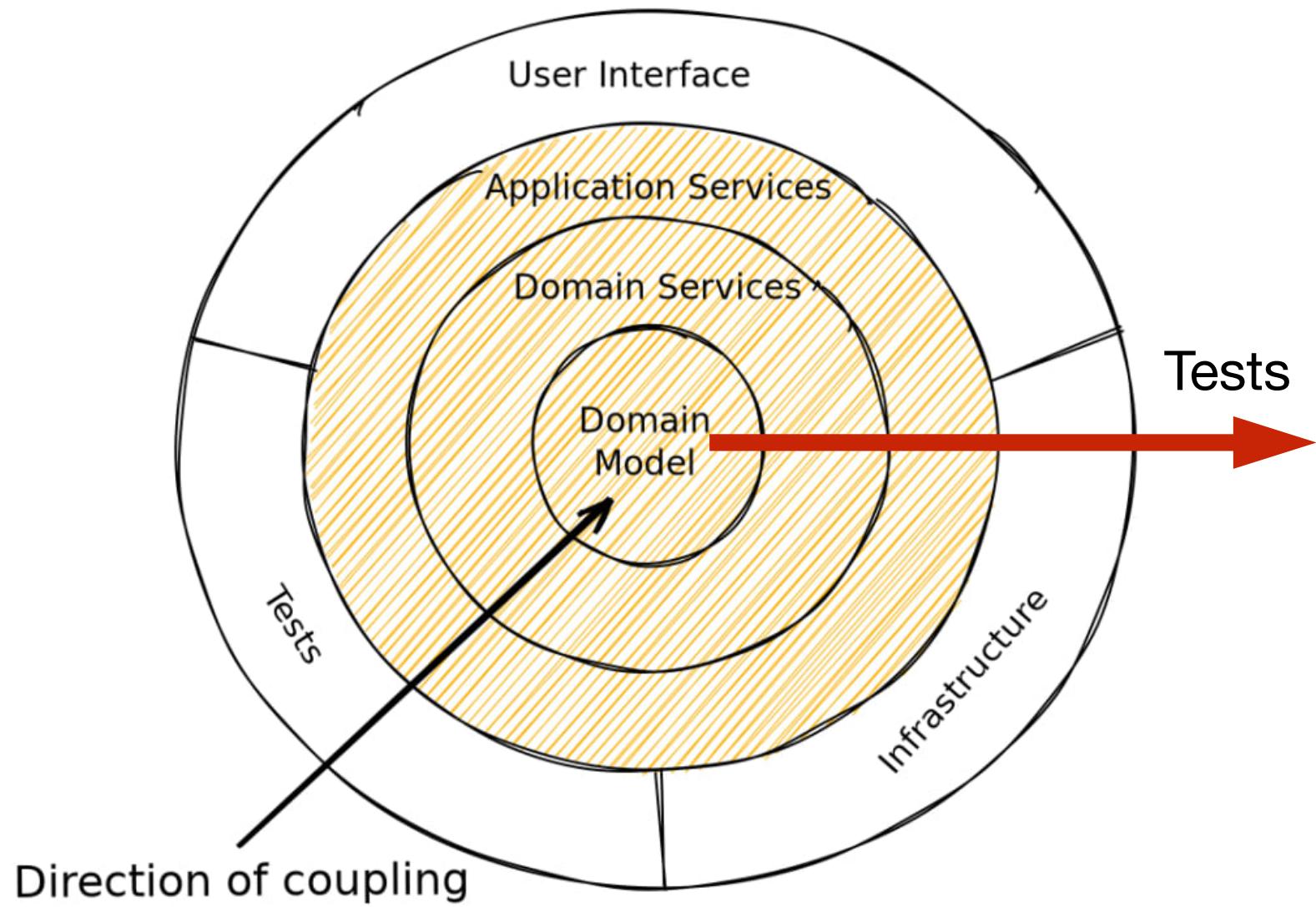


# TCR: Test && Commit || Revert

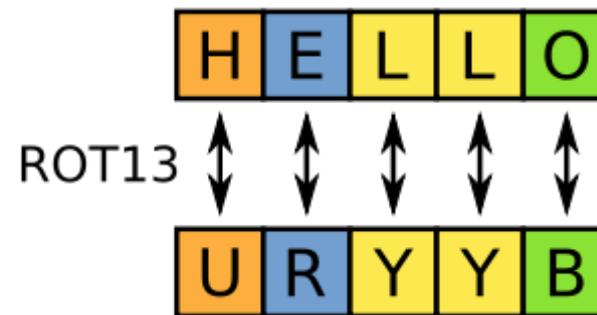
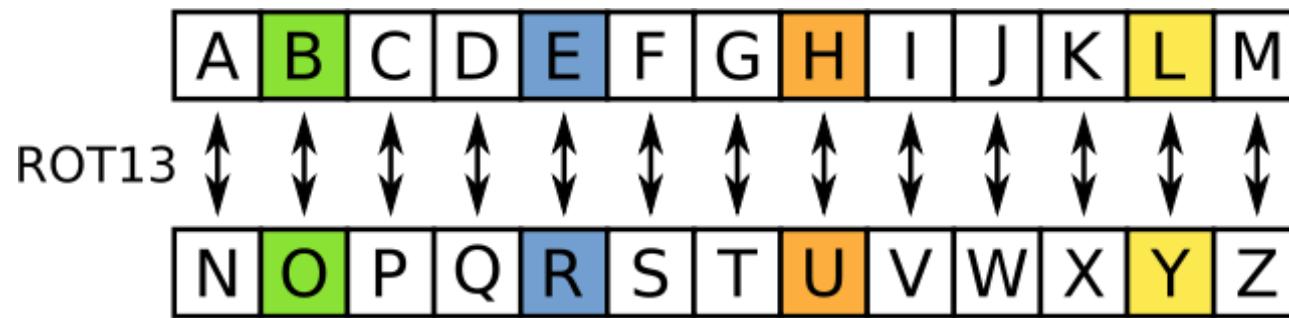


A screenshot of the Visual Studio Code interface showing the settings.json file. The code defines a configuration for Python files, specifically for running tests and committing changes.

```
{  
    "editor.fontSize": 24,  
    "terminal.integrated.fontSize": 24,  
    "editor.lineNumbers": "off",  
    "editor.formatOnSave": true,  
    "python.formatting.provider": "black",  
    "runOnSave.commands": [  
        {  
            "match": ".*\\".py",  
            "command": "clear; python ${file} && git commit -am working || git reset --hard",  
            "runIn": "terminal"  
        }  
    ]  
}
```



1. Interfaz central
2. Cálculos y condicionales
3. Bucles y generalizaciones
4. Casos especiales y manejo de errores





```
Code: github.com/jamesshore/livestream Tag: 2020-05-05 Twitter: @jamesshore Web: jamesshore.com Email: jshore@jamesshore.com

livestream - my_test.js

Testing: myTest.js

1 passing (3ms)

BUILD OK ✓
jamesshore@livestream:~/jshores$ ./build.sh quick
CHECKING Node.js version...
BUILD OK ✓
(0.8%) ✓
jamesshore@livestream:~/jshores$ ./watch.sh quick
CHECKING Node.js version...
BUILD OK ✓
WATCHING arc/**/*
WATCHING arc/**/*


*** BUILD: quick
BUILD OK ✓
(0.8%) ✓
[1]


```

## Ventajas del uso de TDD

- Especificación clara en todo momento. No perdemos el foco ni el objetivo.
- Mejor diseño.
- Confianza, robustez y documentación.
- Ritmo, flujo.

# Críticas

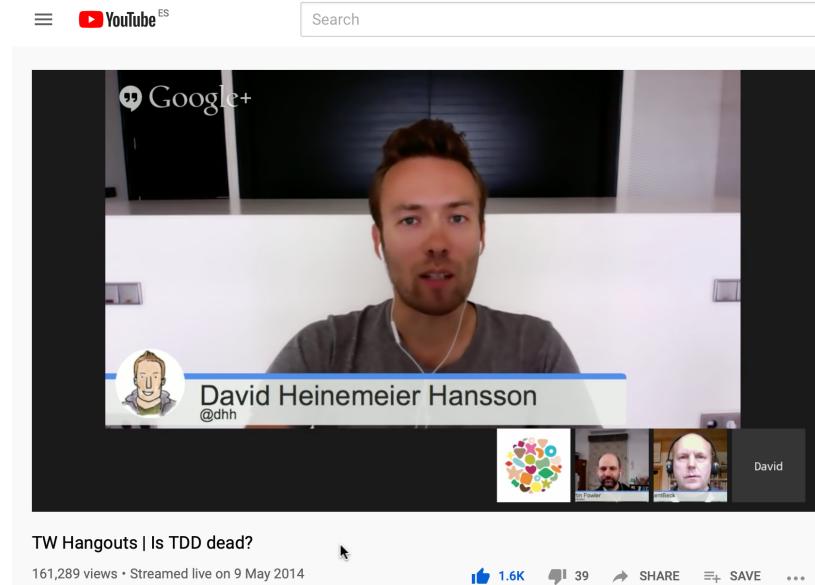
- Lentitud -> ¿comparado con qué?
- Exceso de énfasis en tests unitarios y mocks (Hanson)

**TDD is dead. Long live testing.**

By David Heinemeier Hansson on April 23, 2014

Test-first fundamentalism is like abstinence-only sex ed: An unrealistic, ineffective morality campaign for self-loathing and shaming.

It didn't start out like that. When I first discovered TDD, it was like a courteous invitation to a better world of writing software. A mind hack to get you going with the practice of testing where no testing had happened before. It opened my eyes to the tranquility of a well-tested code base, and the bliss of confidence it grants those making changes to software.



# Ejemplo: dólares y euros

## Cosas por hacer

5 USD + 10 EUR = 10 USD si el cambio es 2:1

5 USD \* 2 = 10 USD

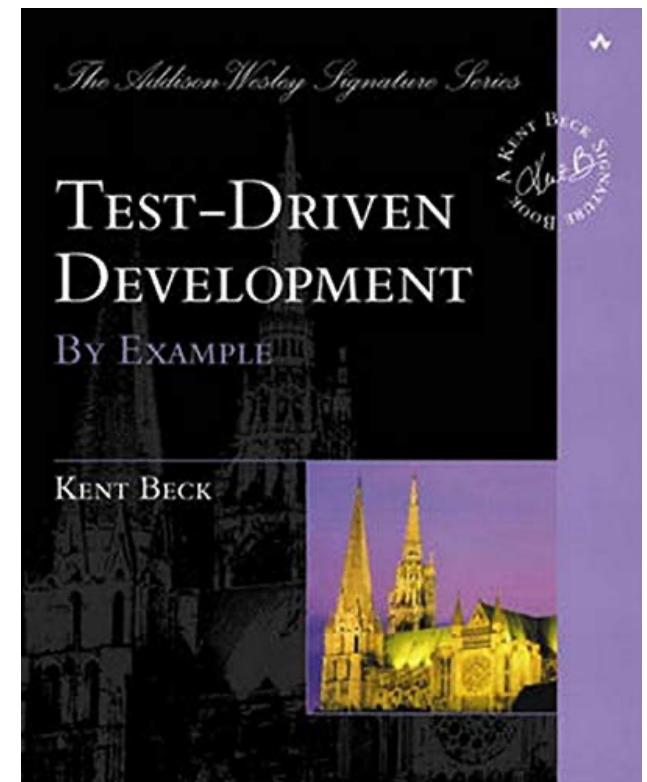
Crear una moneda con 5 dólares

## Test creación de 5 dólares

Vamos a especificar el test para el objetivo de crear una moneda de 5 dólares.

```
import org.junit.Test;
import static org.junit.Assert.assertEquals;

public class TestMonedas {
    @Test
    public void testCrearMoneda() {
        Dolar cinco = new Dolar(5);
        assertEquals(5, cinco.cantidad);
    }
}
```



<https://github.com/dominingogallardo/tdd-monedas>