Test Driven Development

Febrero 2019

Jesús López
jesuslc.com

Sponsors

MARBELLA.
DIGITAL

GDG DevFest
Marbella

eurona

Google

¿Quién soy?

Jesús López

Tech Lead en Qashops (Vente Privée Company)

Ingeniero en Informática

Professional Scrum Master Certified

jesuslc.com
@jeslopcru
github.com/jeslopcru

¿De qué vamos a hablar?

- ¿Por qué hacer Test Driven Development?
- El flujo de TDD
- Herramientas
- ¿Por dónde empezamos?
- Consideraciones

¿Quién hace tests?

Testing

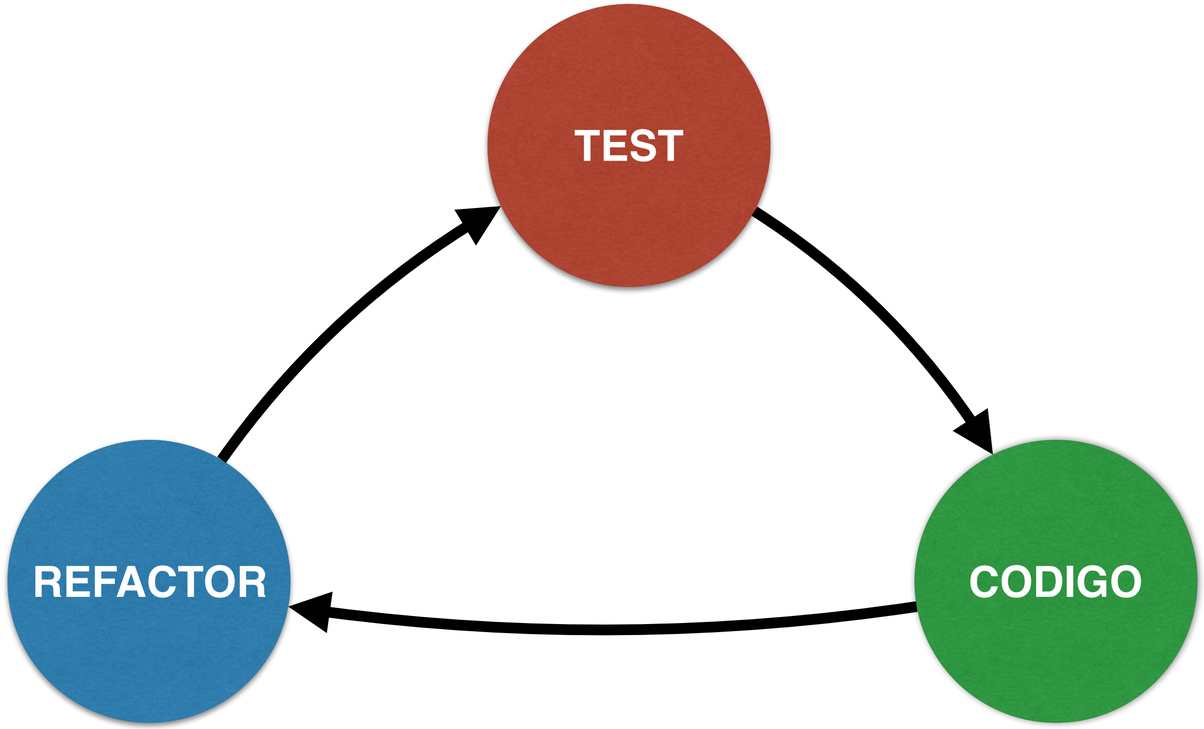
"Testing is a process of executing a program with the intent of finding errors... This definition of testing has many implications... it implies that testing is a destructive process, even sadistic process, which explains why most people find it difficult"

Glendford J. Myers (1979)

Test Driven Development

Escribir un test automático que falla con la funcionalidad que deseamos construir. Después hacer la funcionalidad para que el test funcione y por último refactorizar el código para tener una solución aceptable

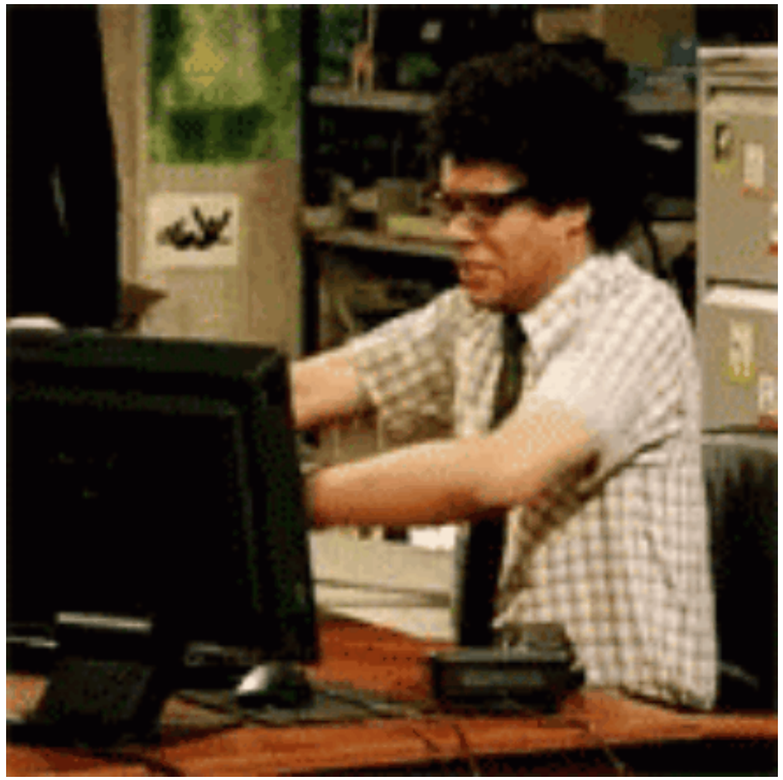
Test Driven Development



Nunca mezclar

¿Por qué hacer TDD?

¿Por qué hacer TDD?



Por qué hacer TDD

- **Hacer software que funciona**
- Hace más estable estable la aplicación
- Tener test nos proporciona un arnés de seguridad
- Sabemos cuando hemos terminado

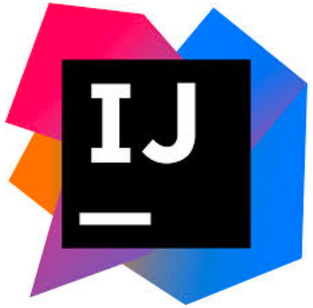
Por qué hacer TDD

- **Hacer software que funciona**
- **menos bugs**
- **paz mental**
- **focus**

Herramientas

Herramientas

Paciencia



¿Por dónde empezamos?

¿Por dónde empezamos?

Katas

¿Por dónde empezamos?
Katas

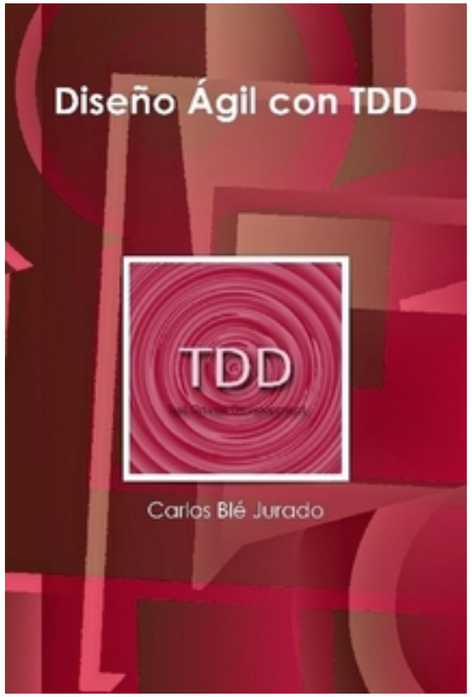
- Ejercicios pequeños
- 30 minutos de duración
- Muchas soluciones distintas

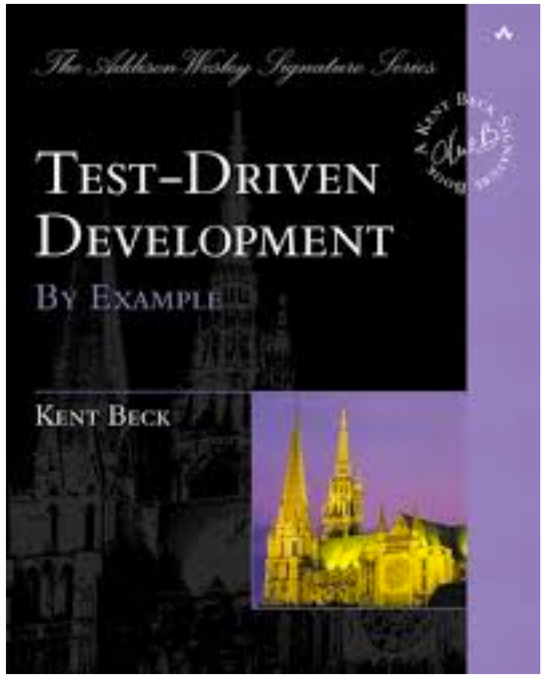
¿Por dónde empezamos?
Katas

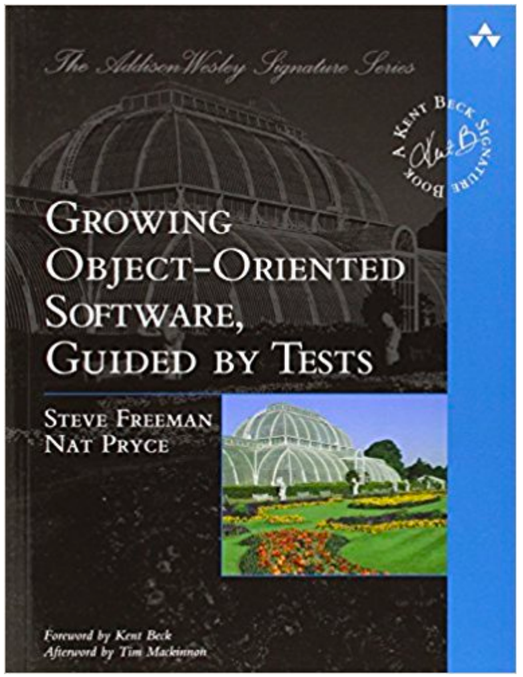
- Ejercicios pequeños
- 30 minutos de duración
- Muchas soluciones distintas

- Las **Katas** son para **aprender**
- Elige un **objetivo**
- Empieza por lo más **sencillo**
- Haz pair programing si puedes

Fuentes recomendadas

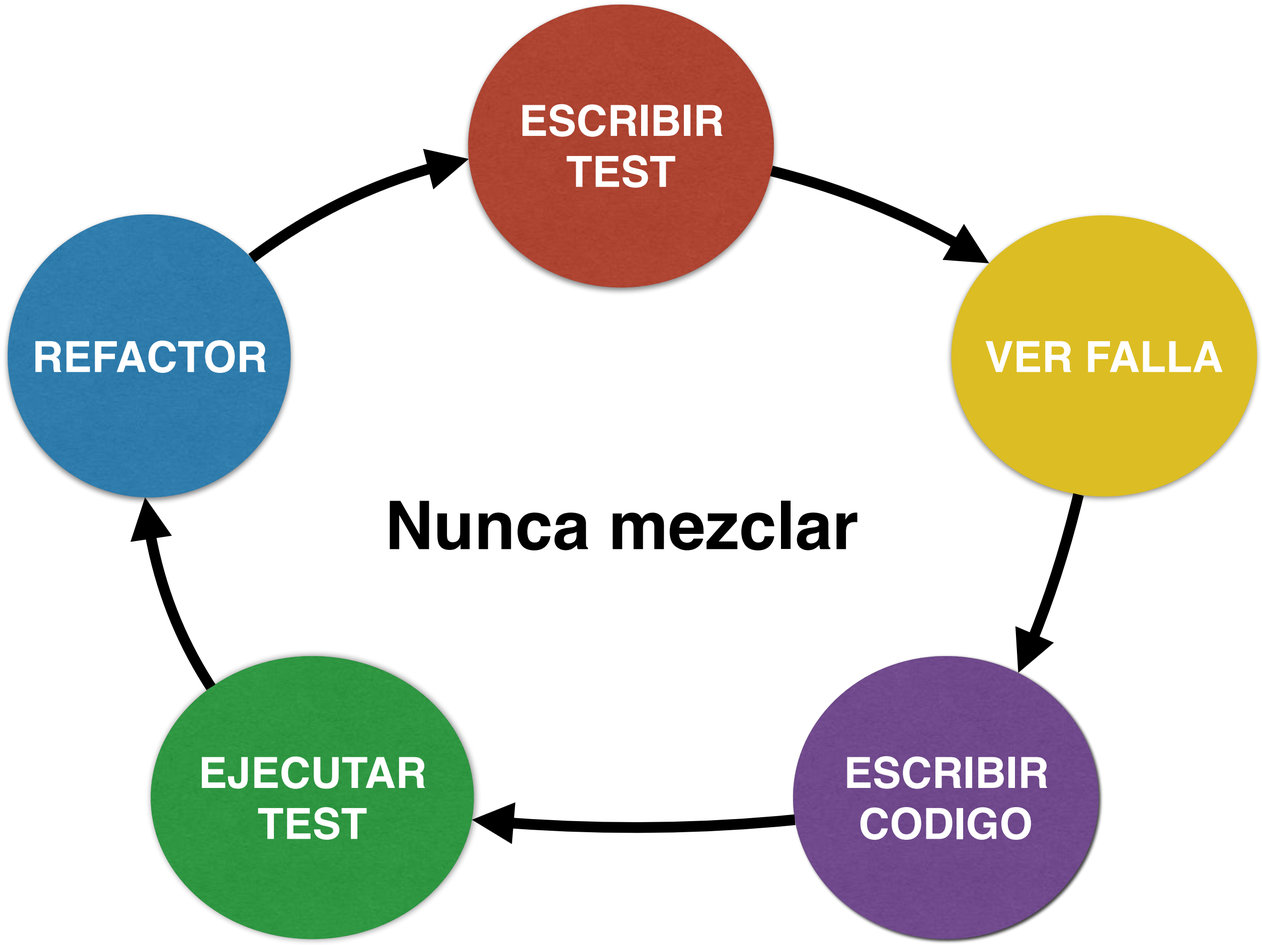




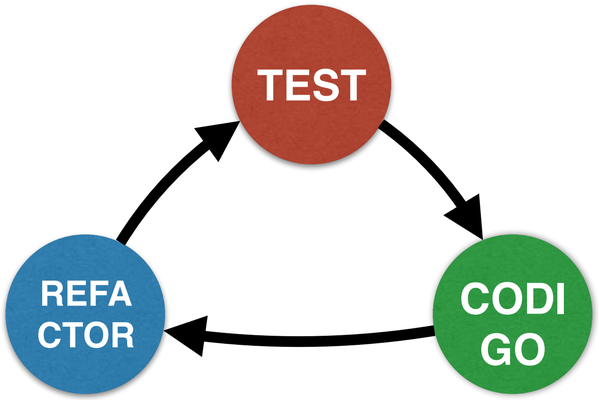


Flujo de TDD

Flujo de TDD


ESCRIBIR TEST
VER FALLA
ESCRIBIR CODIGO
EJECUTAR TEST
REFACTOR
Nunca mezclar

Kata FizzBuzz



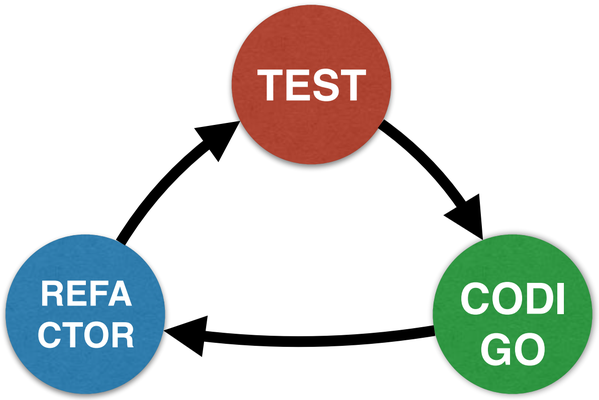
Escribe un programa que imprima los números del **1 al 100**, pero aplicando las siguientes normas:

- Devuelve **Fizz** si el número es divisible por **3**.
- Devuelve **Buzz** si el número es divisible por **5**.
- Devuelve **FizzBuzz** si el número es divisible por **3** y por **5**.

1,2,Fizz,4,Buzz,Fizz,7,8,Fizz,Buzz,11,Fizz,13,14,FizzBuzz,16,17,Fizz, 19,Buzz... etc hasta el 100

https://github.com/jeslopcru/php-bootstrap-kata

Kata DNI



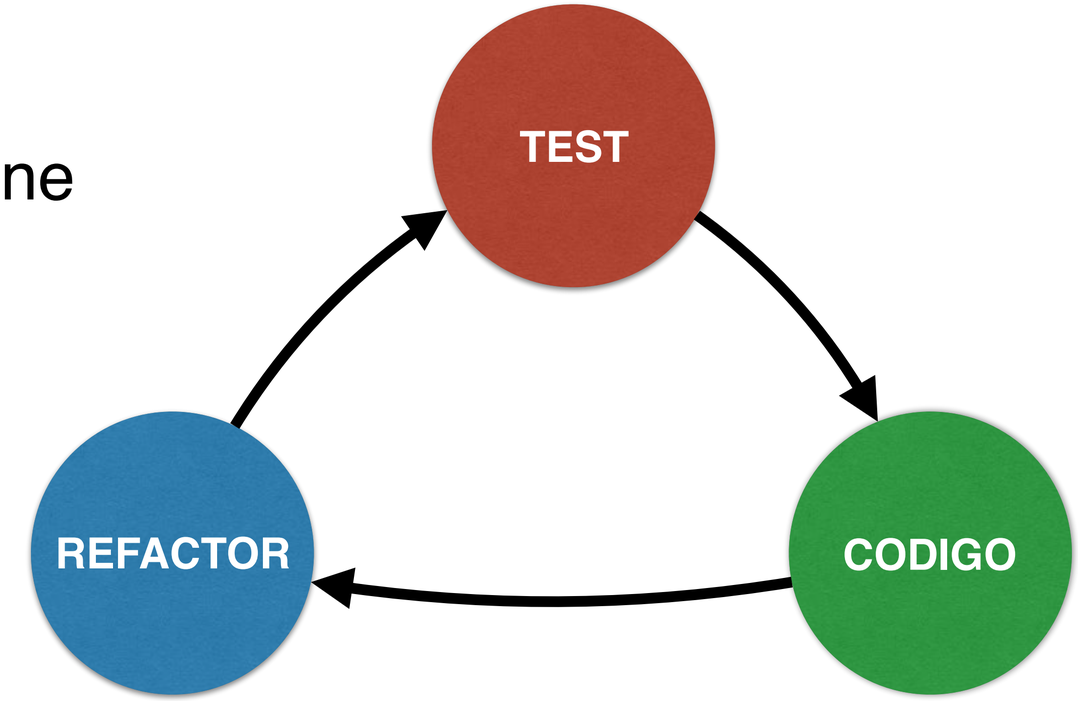
Escribe un programa valide los DNI

Un DNI (Documento Nacional de Identidad) es un identificador que consta de ocho cifras numéricas y una letra que actúa como dígito de control. Existen algunos casos particulares en los que el primer número se sustituye por una letra y ésta, a su vez, por un número para el cómputo de validez que viene a continuación. Este último es el caso del NIE o Número de Identificación para Extranjeros residentes.

https://github.com/jeslopcru/php-bootstrap-kata

En resumen

- Baby steps
- Minima funcionalidad para que funcione
- Refactor en verde
- No perder el foco
- **Hacer katas, no empezar directamente sobre producción**



Gracias

- http://codekata.com/
- https://exercism.io/
- https://rachelcarmena.github.io/2018/11/13/test-driven-programming-workflows.html
- https://github.com/go-to-the-next-level-dev/all-php-katas
- https://github.com/CodiumTeam/php-kata-bootstrap
- https://franiglesias.github.io/

jesuslc.com
@jeslopcru
github.com/jeslopcru

Las 12 reglas

1. Escribir primero un test que falle
2. Mantener los test separados del código
3. Nombres descriptivos de lo que estamos testando
4. Nombre del test describe lo que hace el tests
5. 1 y solo 1 razón para fallar
6. Escribe la aserción primero (de abajo arriba)
7. Los nombres deben comunicar la intención
8. Arregla el código cada vez que esté roto
9. Escribe el código más simple que lo arregle
10. Asegurate que la aserción falle
11. Una vez que el test esté en verde refactoriza
12. Une los test duplicados y parametrizalos