

# Test Driven Development

PHPSevilla - Noviembre 2017

Jesús López  
[jesuslc.com](http://jesuslc.com)

# La comunidad

**PHP**Sevilla

[www.phpsevilla.com](http://www.phpsevilla.com)



# Sponsors



ESPACIO  
DE RESILIENCIA  
CREATIVA





# Comunidad



fe(x)t | what's coming  
next festival



SVQXDG

Sevilla Xamarin Developer Group



NS



# Test Driven Development

**Jesús López**

Ingeniero en Informática

Software Engineer

Professional Scrum Master & ITIL Certified

**jesuslc.com**

@jeslopcru

github.com/jeslopcru

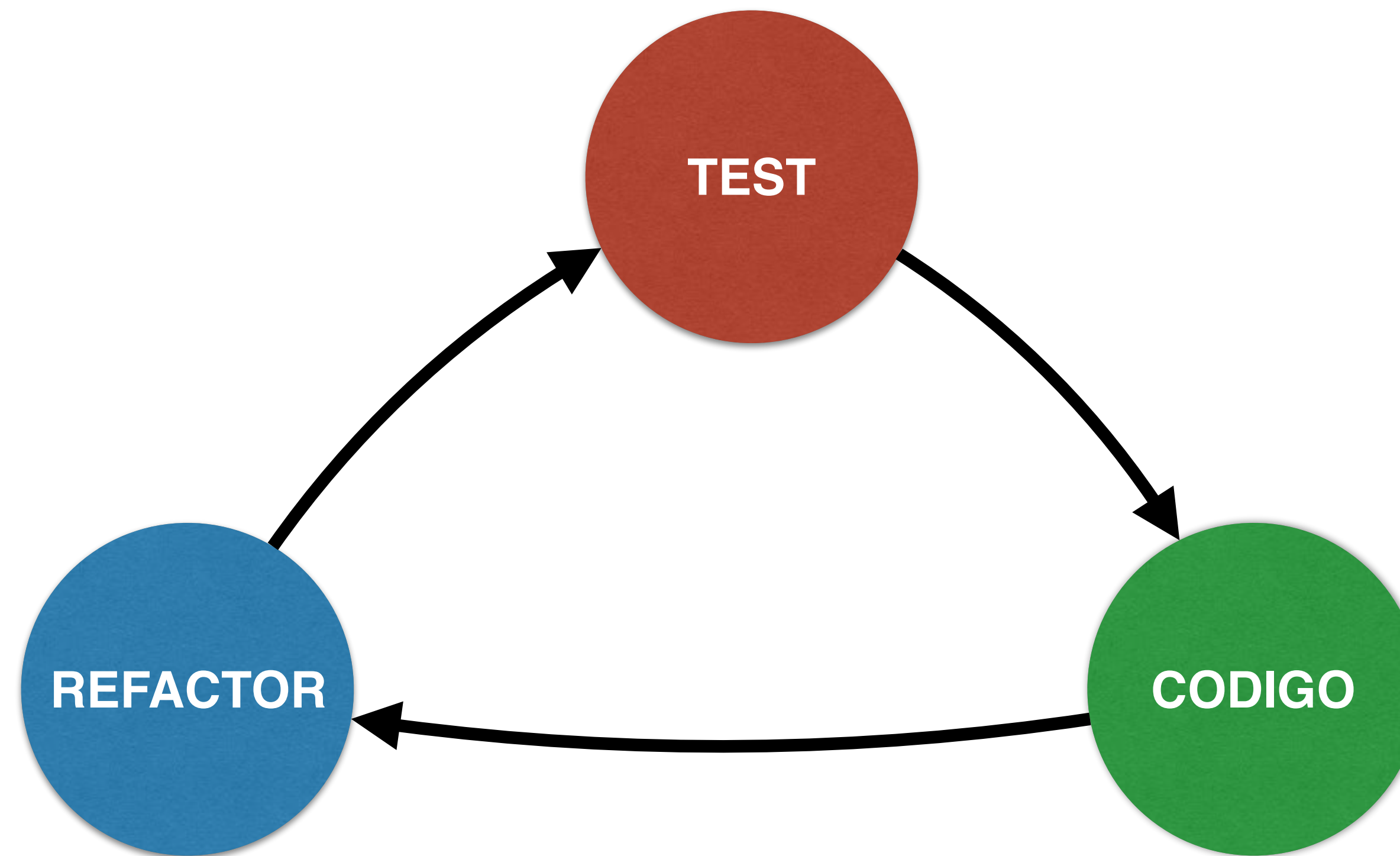
# Test Driven Development

"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)



# ¿Qué hacemos?



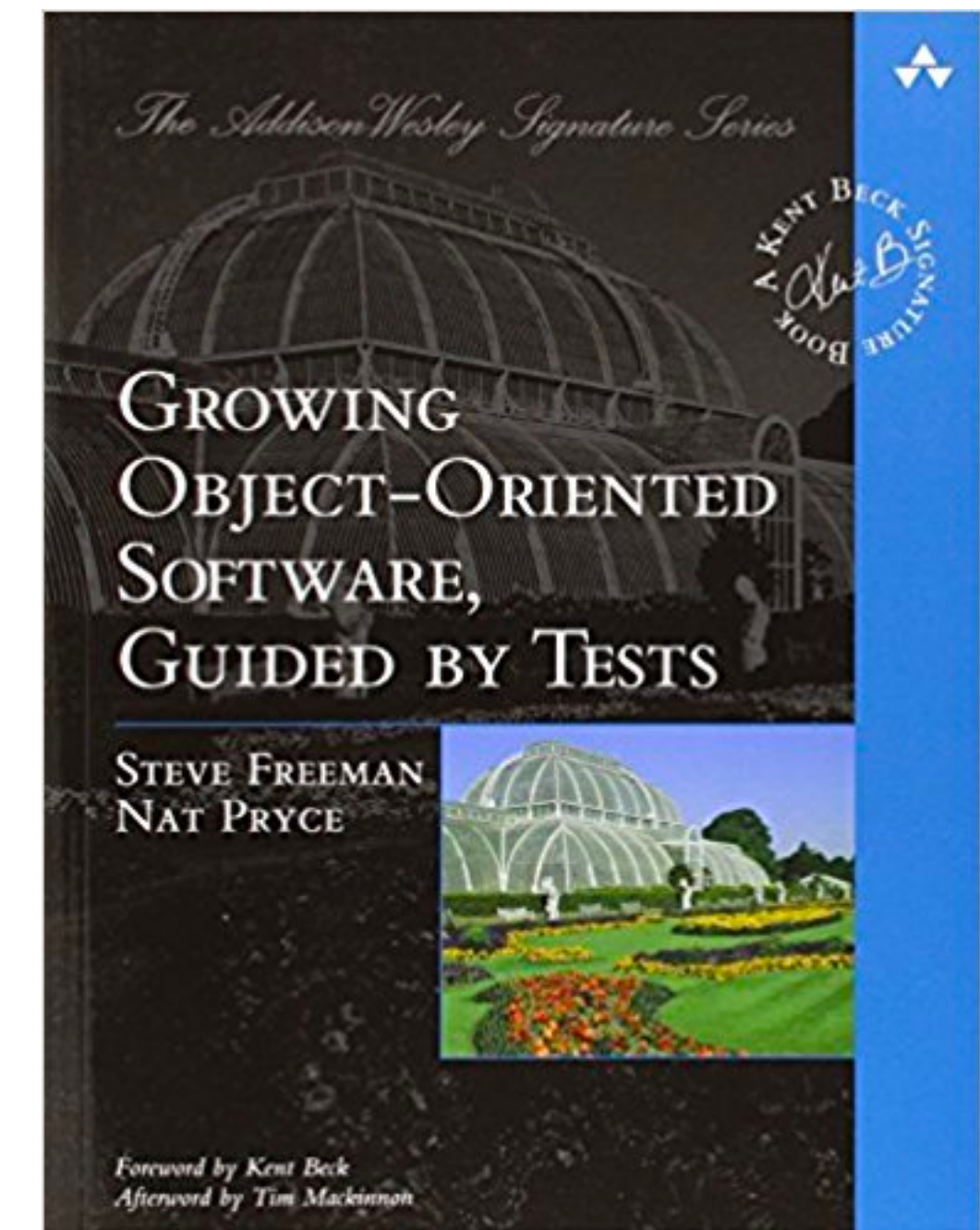
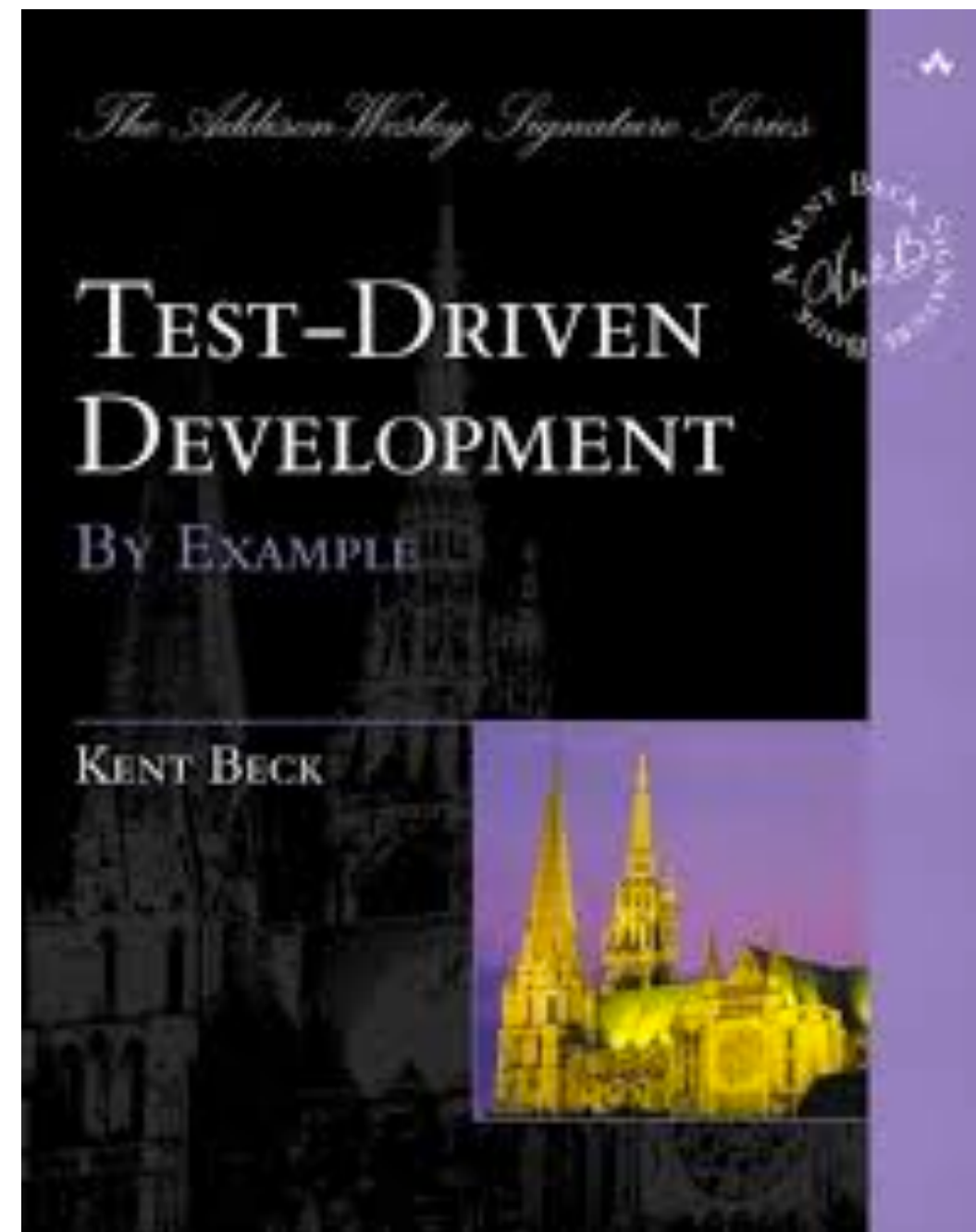
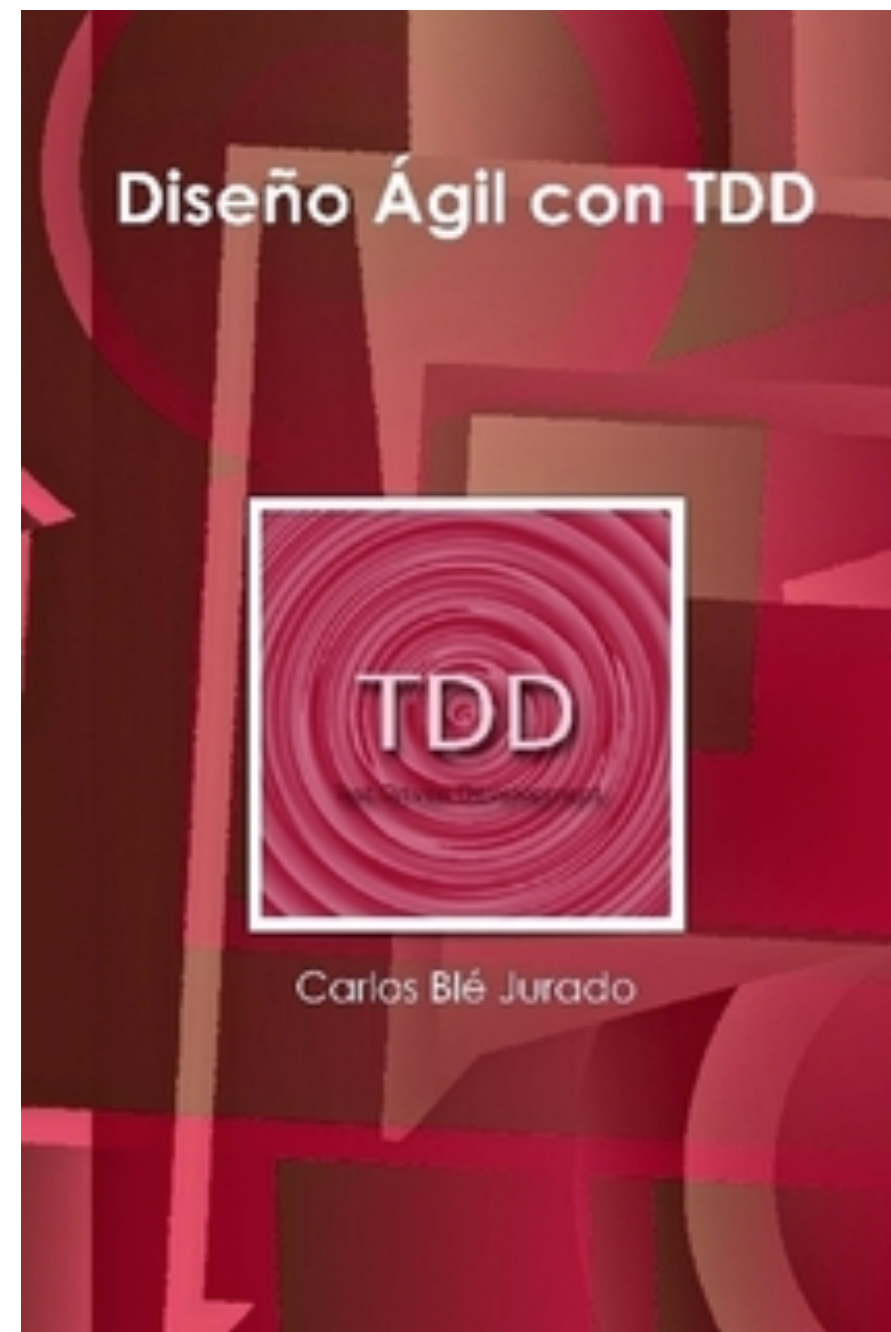
**Nunca mezclar**

# 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



# Fuentes recomendadas



# ¿Por dónde empezamos?

## 守破離

Shuhari



# ¿Por dónde empezamos?

## 守破離

★ shu (守?) "Proteger", "obedecer"

★ ha (破?) "desapego", "desprendimiento"

★ ri (離?) "Dejar", "separar", "trascender"



# ¿Por dónde empezamos?

## Katas

- ★ Las katas son para **aprender**
- ★ Elige un **objetivo**
- ★ Ejercicio previo de análisis
- ★ Empieza por lo **sencillo**
- ★ Haz pair programming si puedes



Baby  
Steps

# ¿Por dónde empezamos?

## Katas

### Baby Steps

- ★ Las katas son para **aprender** → No es necesario terminar
- ★ Elige un **objetivo** → Follow rules
- ★ Ejercicio previo de análisis → Lista de tests
- ★ Empieza por lo **sencillo** → Paso a paso
- ★ Haz pair programming si puedes → No acapares el teclado

# Herramientas

- IDE
- Git
- Tests
- **Paciencia**

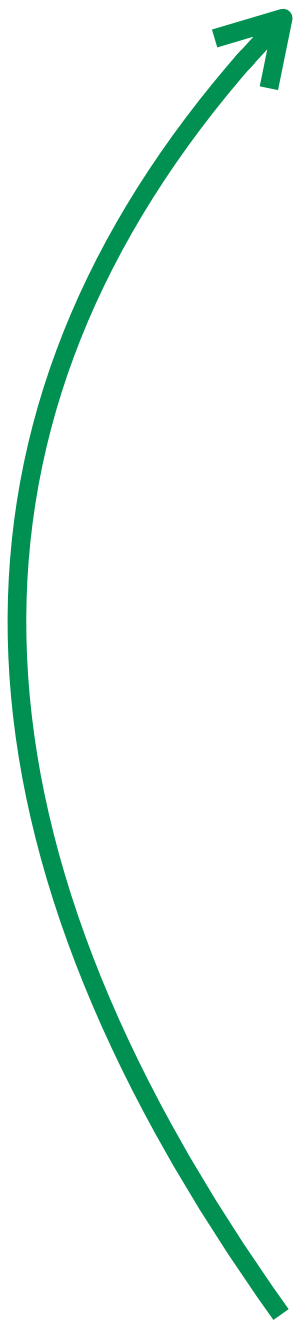
Sin test no hay  
código

Commits  
pequeños

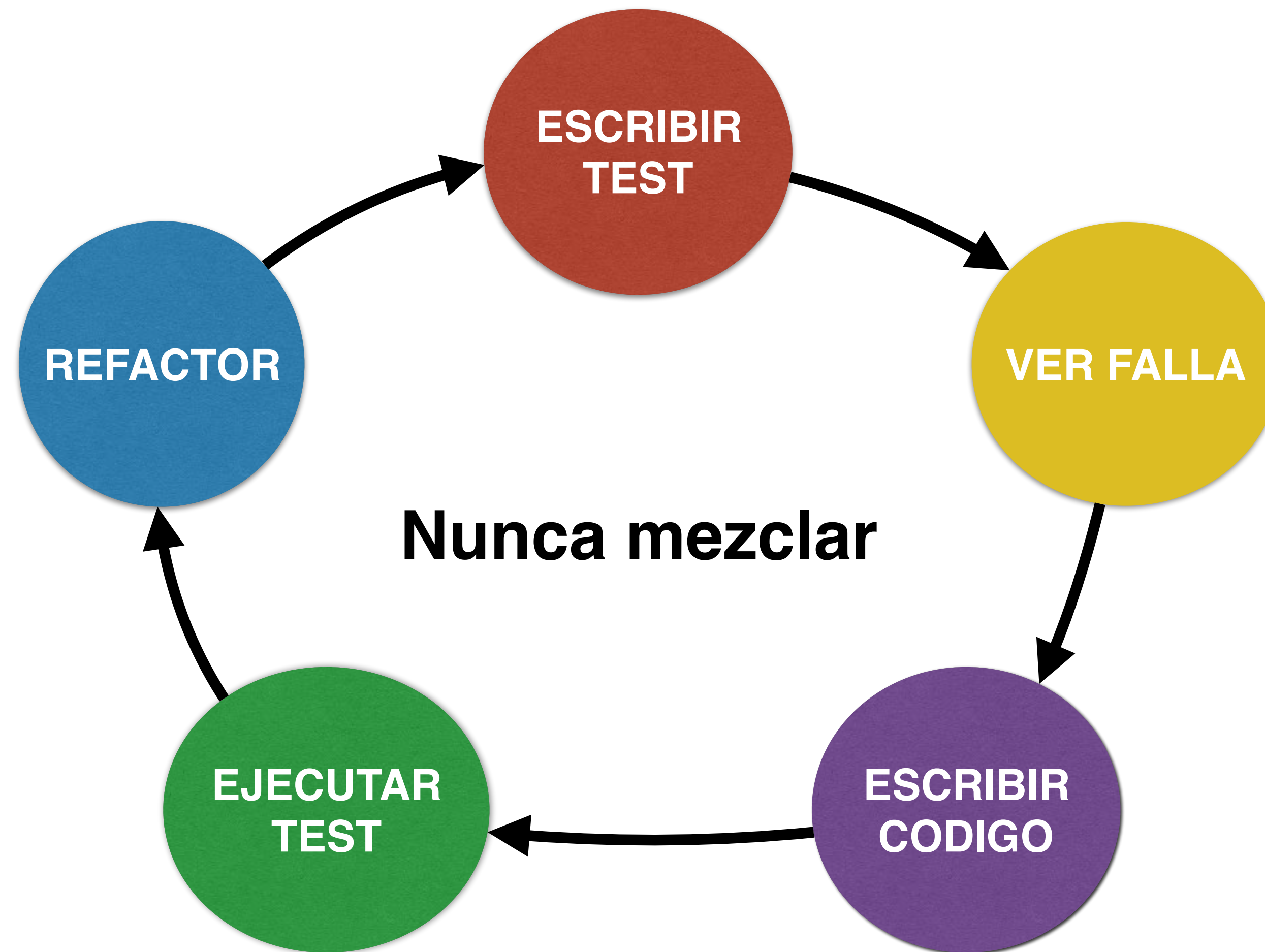
Test en Verde  
para refactorizar



# Pasos para TDD

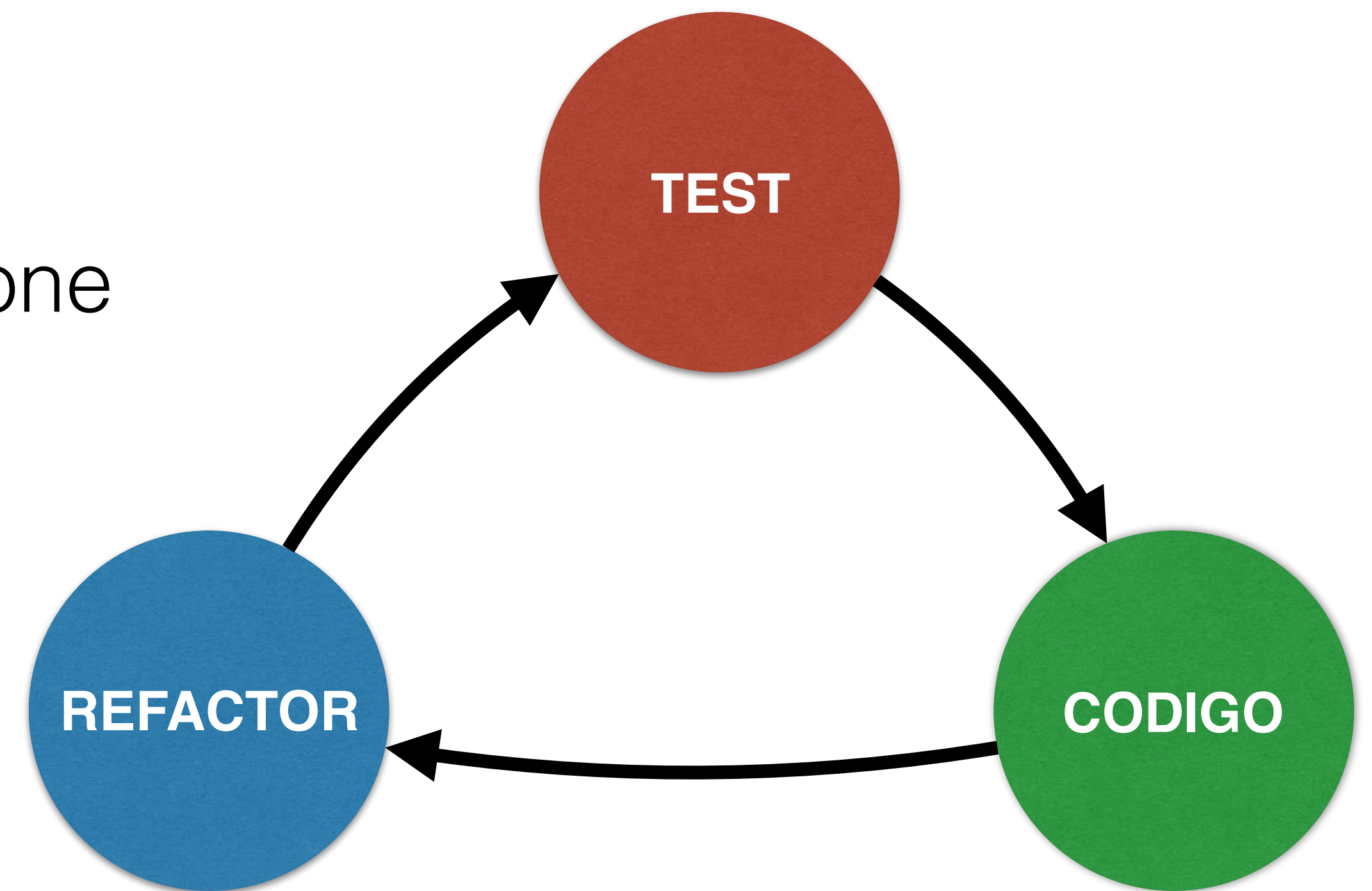
1. Lista de tests
  2. Escoger un test de la lista
  3. Escribir **Test** que falle
  4. **Mínima funcionalidad** para pasar el test
  5. Comprobar los test
  6. Refactor: **¡FUNCIONA!**
- 

# Pasos para TDD



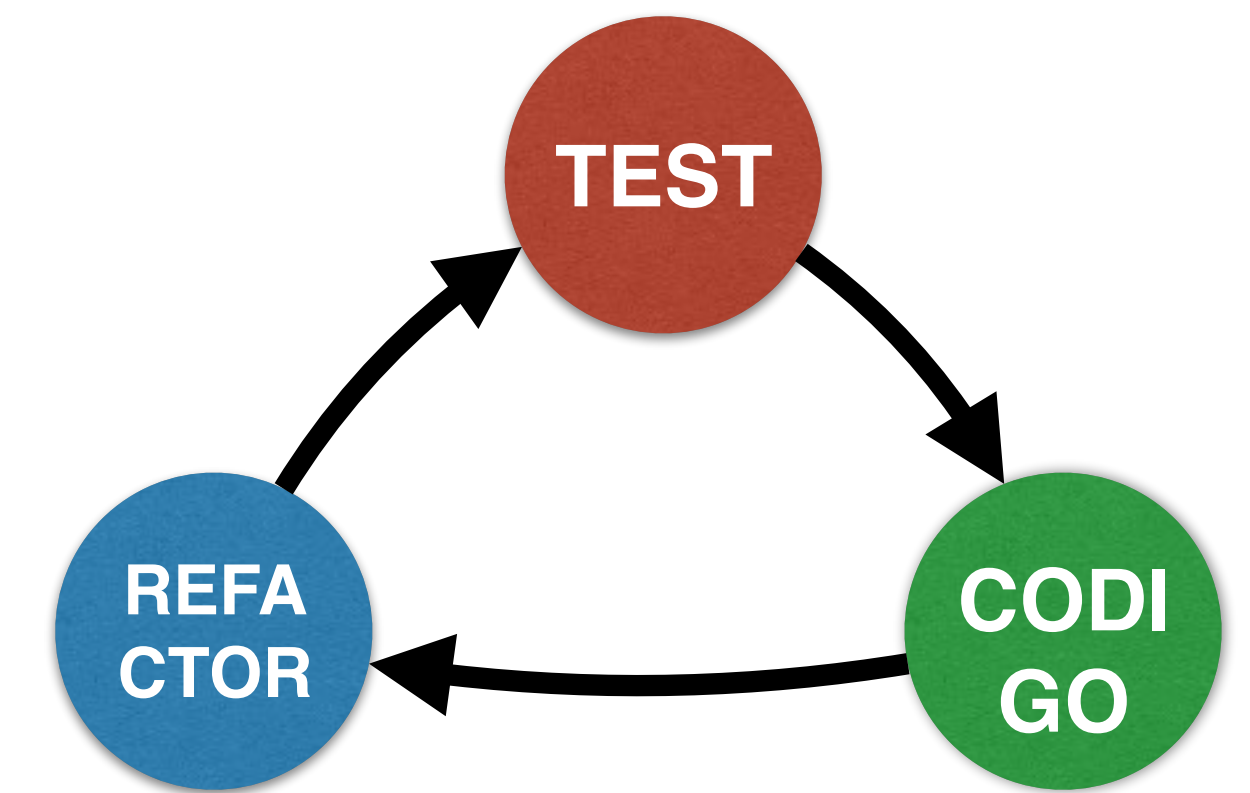
# En resumen

- Baby steps
- Minima funcionalidad para que funcione
- Refactor en verde
- Una tarea cada vez
- Iterar





# 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**.

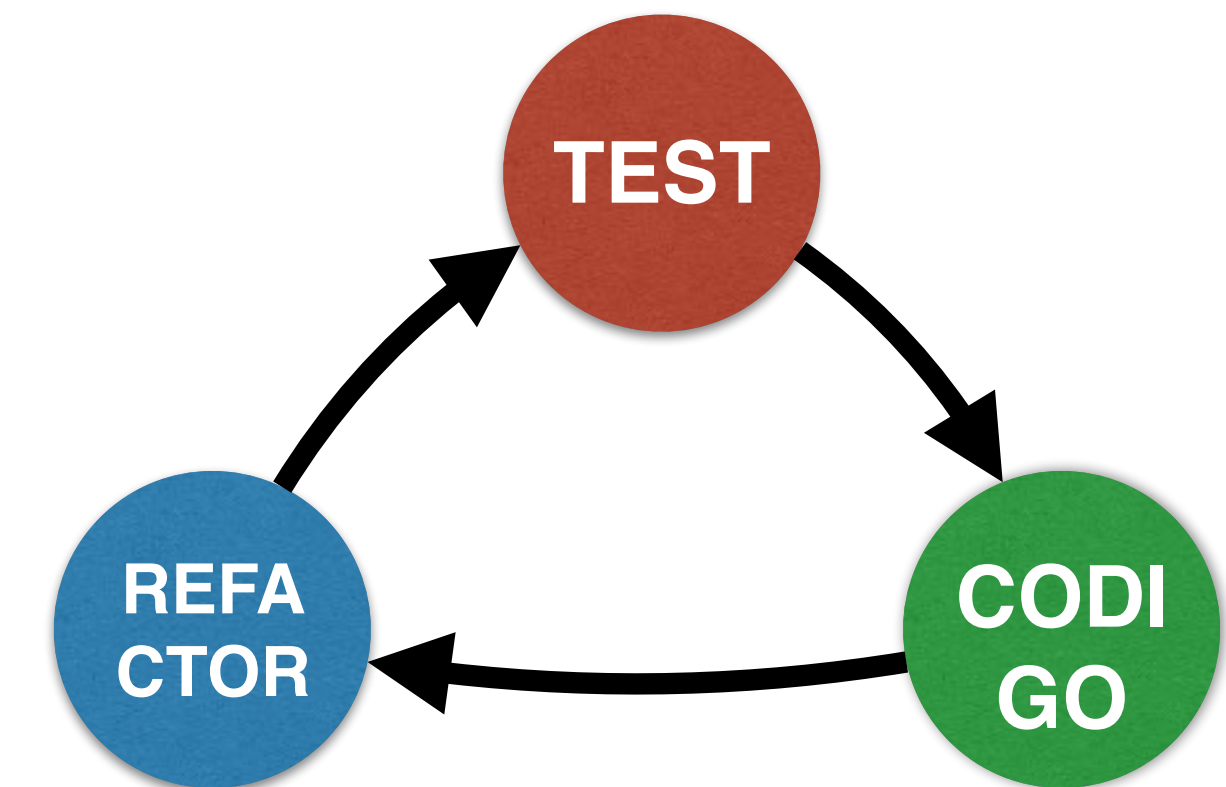
*1,2,Fizz,4,Buzz,Fizz,7,8,Fizz,Buzz,11,Fizz,13,14,FizzBuzz,16,17,Fizz,19,Buzz... etc hasta el 100*

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

# Kata Password Validator

We want to ensure that our users' passwords have the following rules:

- Have more than 8 characters
- Contains a capital letter
- Contains a lowercase
- Contains a number
- Contains an underscore



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