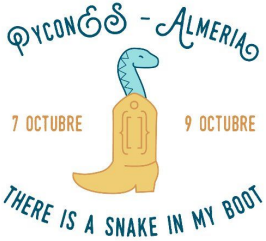


Introducción a TDD

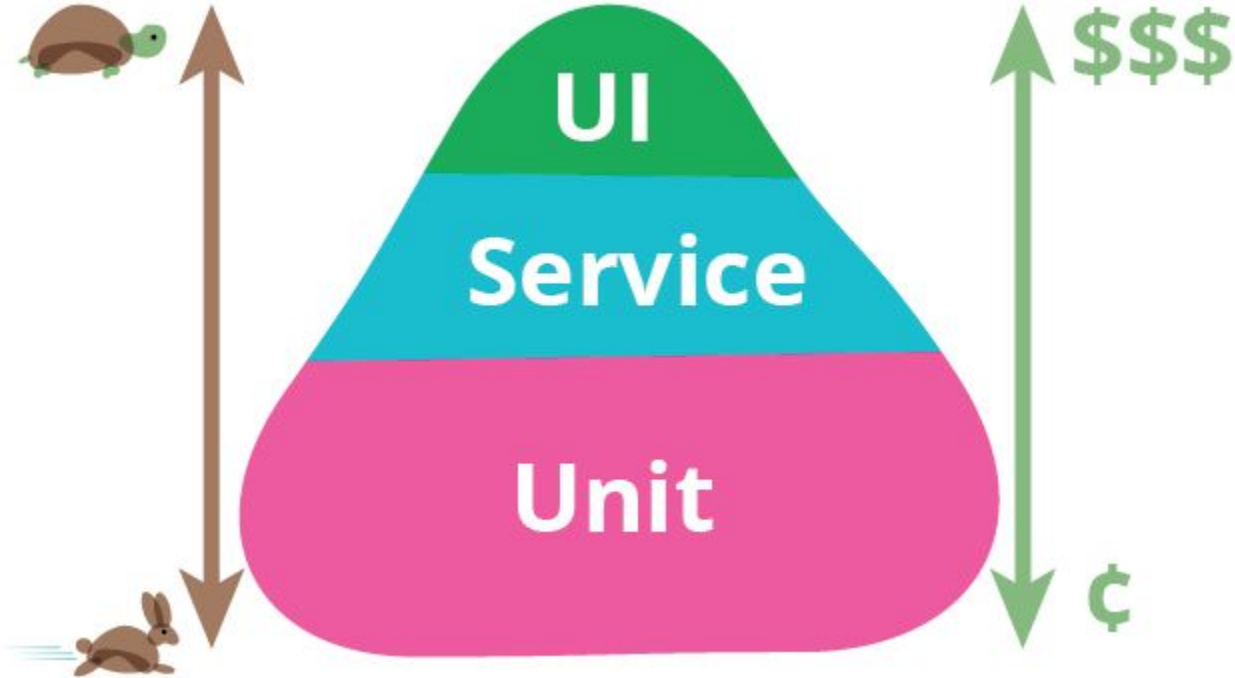
PyConEs 2016

https://github.com/aleasoluciones/bifer

Para qué estamos aquí

- Para divertirnos, aprender y compartir
- Qué es TDD (Test-Driven Development)
- Qué es una kata
- Trabajar en parejas y con pomodoros

Test Pyramid



Fuente: http://martinfowler.com/bliki/TestPyramid.html

¿Qué es un test unitario?

Fast

Isolated/**I**ndependent

Repeatable

Self-Validating

Thorough and Timely

https://github.com/ghsukumar/SFDC_Best_Practices/wiki/F.I.R.S.T-Principles-of-Unit-Testing

https://pragprog.com/magazines/2012-01/unit-tests-are-first

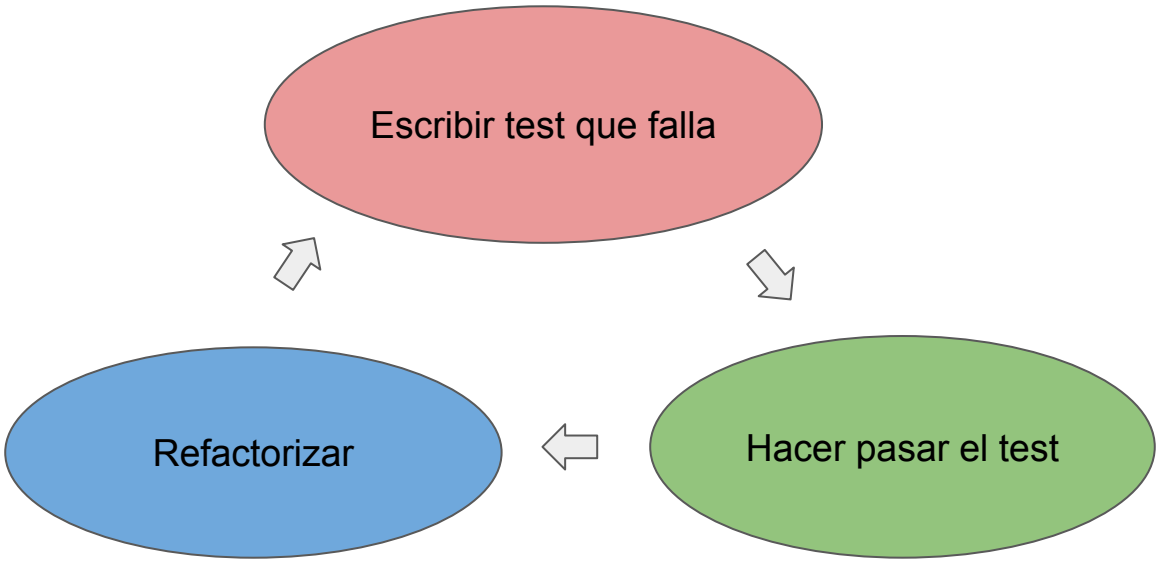
Estructura de un test

```
with describe(‘Mi funcionalidad de suma’):
    with context(‘cuando sumo 2 y 2’):
        with it(‘el sistema retorna 4’’):
            calculadora = MiCalculadora()          # Arrange
            resultado = calculadora.suma(2, 2)     # Act
            expect(resultado).to(equal(4))         # Assert
```

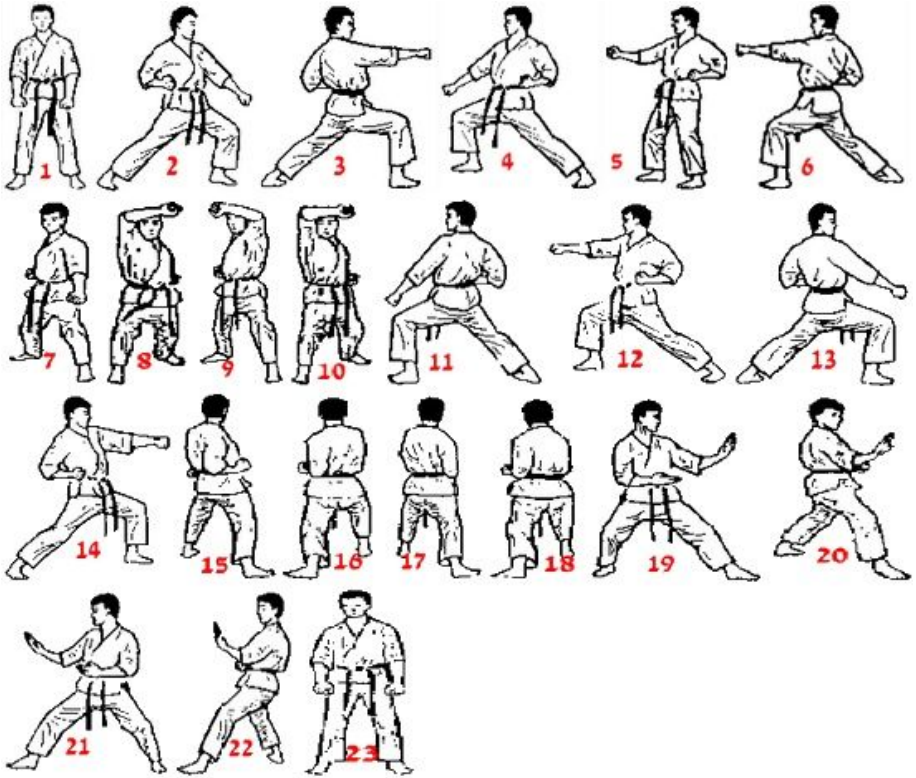
¿Qué es TDD?

- Técnica de desarrollo de software
- Ayuda a:
  - **Enfocarte** en la funcionalidad a desarrollar: escribir menos código
  - **Descubrir** poco a poco cómo resolver el problema
  - Crear mejores **tests** de tu código
  - Código de producción cubierto por tests (**red de seguridad**)
- **No** necesariamente ayuda a mejorar tu diseño

Ciclo de vida de TDD



¿Qué es una kata?



http://codekata.com/

Herramientas a utilizar

- mamba
  - test runner con filosofía BDD: contextos y qué hace el sistema
- expects
  - ante una acción, ¿qué espero que ocurra?
  - expect(resultado_real).to(resultado_esperado)
- Ejemplo: https://github.com/aleasoluciones/pycones2016

Kata FizzBuzz

http://www.solveet.com/exercises/Kata-FizzBuzz/11

https://github.com/aleasoluciones/pycones2016

Cómo vamos a hacerlo:

- **Pomodoro** (3 iteraciones de 25 minutos cada una)
- **Pair programming** + Ping Pong
- Escribe lo mínimo para que funcione y pase el test
- **Mob Programming**: ¿os hace?

Kata FizzBuzz: formato iteraciones

A tirar líneas (17 minutos)

8 minutos para feedback

- show me the code
- ¿dudas?
- ¿cambio de parejas?

Bola extra

- Devolver “**Woof**” si un número es divisible por 7
- Devolver “Fizz” si es divisible por 3 **o si incluye** un 3 en el número
- Devolver “Buzz” si es divisible por 5 **o si incluye** un 5 en el número
- No usar **else**
- Un **único nivel de indentación** por método
- No encadenar métodos: ley de Demeter (principio del menor conocimiento)
- Sin usar **if()**

Siguientes pasos

- Ideas para katas: https://github.com/12meses12katas
- Bibliografía
  - TDD by example (Kent Beck)
  - Growing Object-Oriented guided by tests (Steve Freeman, Nat Pryce)
  - Diseño ágil con TDD (Carlos Blé)
- Is TDD dead?
  - http://martinfowler.com/articles/is-tdd-dead/
  - https://www.youtube.com/watch?v=z9quxZsLcfo

Screencasts

- Screencasts de Sandro Mancuso:
  - https://www.youtube.com/watch?v=iZjgj1S0FCY
  - https://www.youtube.com/watch?v=XHnuMjah6ps

- Carlos Blé
  - Implementando algoritmos con TDD
  - Kata de las cartas

Ya casi nos vamos…

Feedback del taller:

- Sensaciones generales
- ¿Qué os ha parecido TDD?
  - ¿Fácil, difícil?
  - ¿Valioso, inútil?
- ¿Alguien se plantea empezar a probarlo en su día a día?

Feedback visual: Niko-niko

Fuente: https://github.com/starmer/team-mood-tracker


That's all Folks!