

# ¿Está muerto TDD?

Máster en Cloud Apps

Desarrollo y despliegue de aplicaciones en la nube

Autor: Daniel Sánchez González

Tutor: Luis Fernández Muñoz



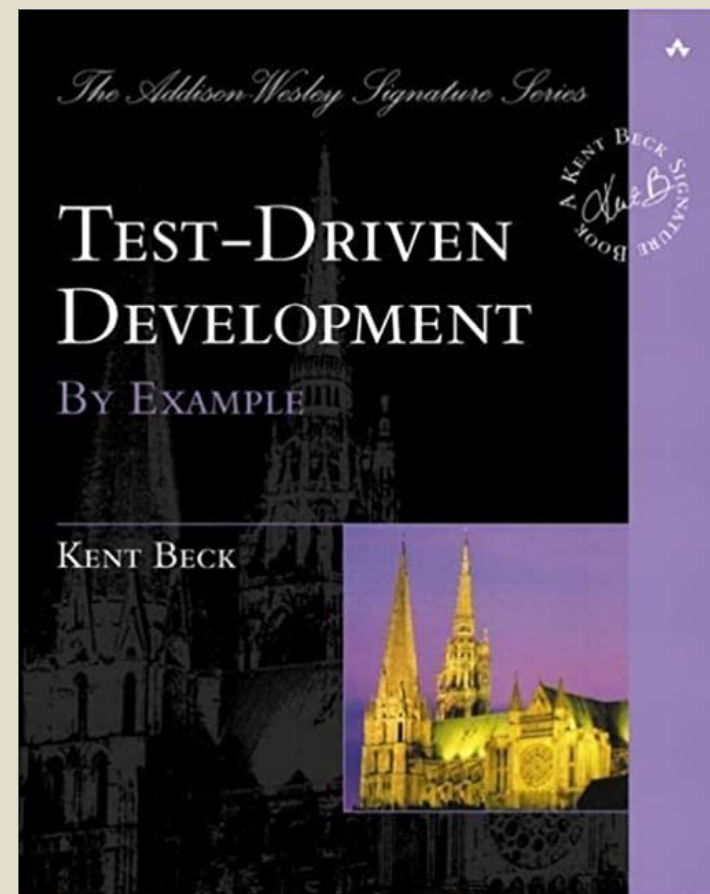
Creative Commons Attribution-ShareAlike license  
<https://creativecommons.org/licenses/by-sa/4.0/>

# INTRODUCCIÓN



# Introducción

- ¿Está muerto TDD?
- Debate 2014
  - DHH, Fowler, Kent Beck
- Libro 2002
- Contexto del máster



David Heinemeier




Martin Fowler



Kent Beck





# Introducción >> Contenido

1. Breve descripción

2. Debate

3. Ejemplos

4. Conclusiones

# BREVE DESCRIPCIÓN



# 1. Breve descripción

Red

- Demostrar ausencia

Green

- Entender solución

Refactor

- Eliminar duplicación

Red

Green

Refactor

# DEBATE



## 2. Debate

Diseño emergente

Chicago VS London

Velocidad

Grado aislamiento

Tipos de pruebas



## 2. Debate >> Diseño emergente

- Uso de metáforas
- Pensar en lo que debe hacer el sistema y dejar que aflore



## 2. Debate >> Diseño emergente



Jim Coplien

- Hacerlo bien = TDD
- Proyectos sin arquitectura y con TDD
- Kent Beck: Hagan arquitectura. No se dejen la piel



David Heinemeier

- Fe basada en TDD



Bertrand Meyer

- Visión de túnel
- No es para tomárselo en serio
- Es necesaria una visión de alto nivel



## 2. Debate >> Diseño emergente

- Error de la comunidad (metáforas)
- **Visión arquitectónica previa**
- **Emerge el diseño cercano al código**



Uncle Bob

- Refactorización != herramienta diseño
- Emerge el micro-diseño
- **Hay que saber diseñar**



Sandro Mancuso



## 2. Debate >> Diseño emergente



¿Grado de diseño previo?

## 2. Debate >> Chicago VS London



### CHICAGO

- Inside-out
- Caja negra
- Destreza en fixtures
- Modelo de negocio



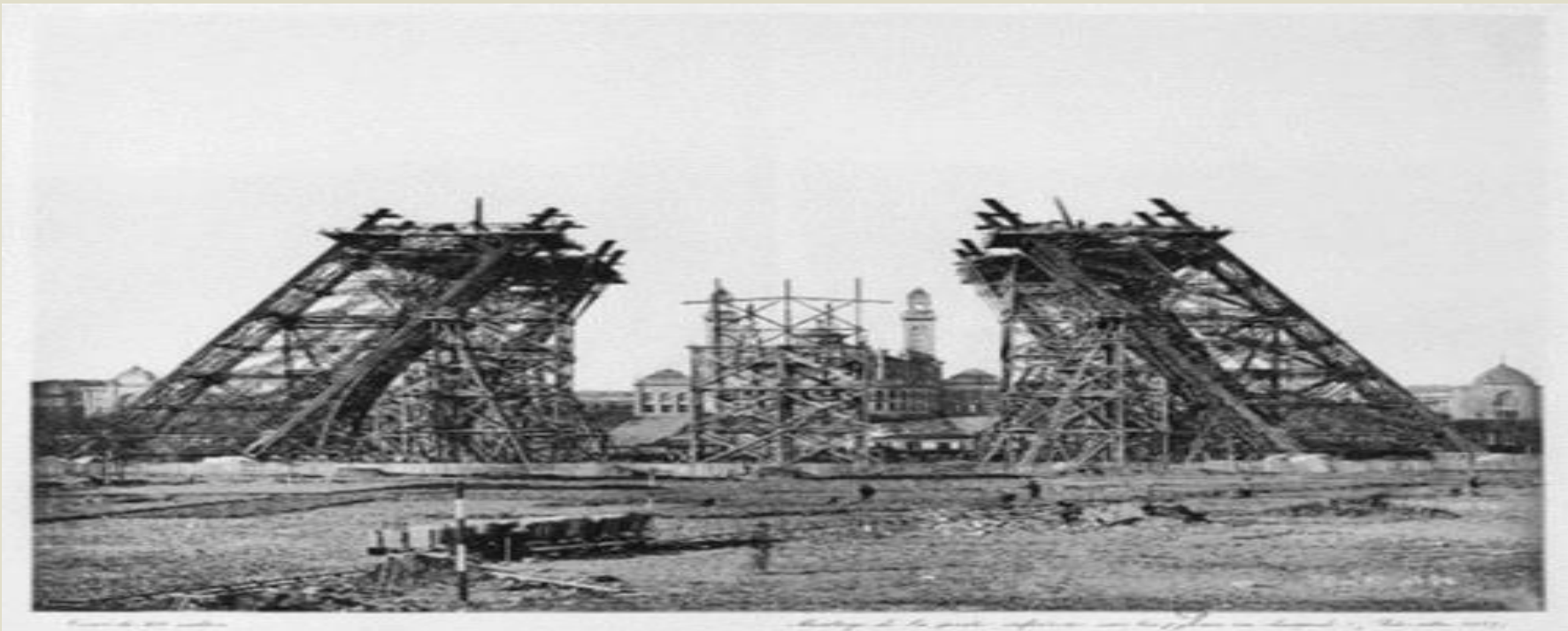
### LONDON

- Outside-in
- ATDD + Mocks
- Destreza en mocks
- Orientación usuario (API, UI)



## 2.1 Debate >> Chicago VS London

- Known-to-unknown
- Contradicciones
- Evitar mocks refactorizando
- Modelo "Eiffel"





## 2. Debate >> Chicago VS London



Ian Cooper

- Mocks = caja blanca (acople + fragilidad)
- TDD clásico = caja negra

- Chicago más lento
- Métodos compatibles
- Mocks = diseñar límites sistema



Sandro Mancuso

## 2. Debate >> Chicago VS London



¿Elección de estilo?

## 2. Debate >> Velocidad

- Metáfora del pozo
- Reducción del miedo (mínimos tests)
- Granularidad: # tests / decisión diseño
- Avance hacia exterior sistema
- Obvious implementation
- Gears

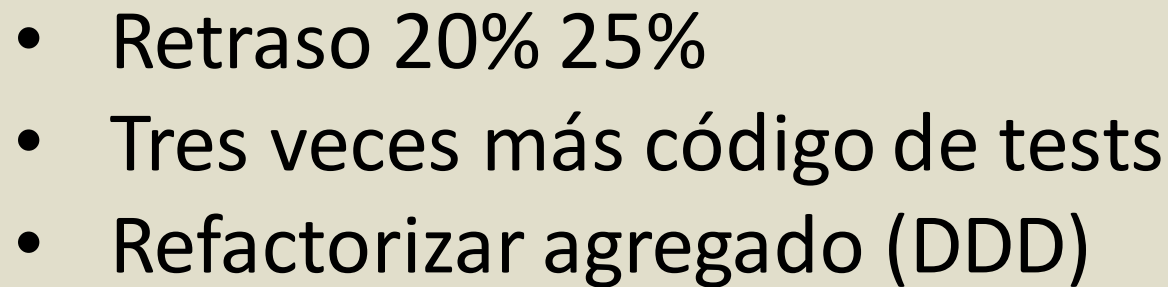




## A vibrant, abstract tapestry by Kiki Karamouzos, featuring a dense, colorful pattern of geometric shapes, swirling lines, and organic forms in shades of red, blue, green, yellow, and purple. The design is highly detailed and layered, with various motifs including triangles, spirals, and wavy lines. The overall effect is a rich, textured composition that changes as the viewer's perspective shifts.



- Estudio Microsoft
  - Aumento tiempo: 15%-35%
  - Cobertura: 79% - 88%



## 2. Debate >> Velocidad



¿Granularidad?



## 2. Debate >> Grado de aislamiento

- Aislar problemas. I/O claras
- Trade off: TDD no es responsable del diseño
- Testeo deficiente => ¿mal diseño?





## 2. Debate >> Grado de aislamiento



David Heinemeier

- Diseño forzado con indirecciones
- Búsqueda de Unit tests sin colaboradores y rápidos
- Arquitectura dañada: dirigida por mocks

## 2. Debate >> Grado de aislamiento

- Aislar los tests, no el código
- La gente entendió mal y usa mocks
- Evitar compartir datos en BD



Ian Cooper

- TDD no prescribe grado aislamiento
- Se persigue el aislamiento para probar más



Martin Fowler

- Testeabilidad y DIP
- TDD orienta mejor a principios de diseño
- Problema indirecciones y dificultad



Emily Bache

## 2. Debate >> Grado de aislamiento



¿Grado de indirección?



## 2. Debate >> Tipos de pruebas

- ATDD:
  - Problema técnico (fixtures)
  - Problema social: stakeholders
  - Tiempo largo de feedback
- Contradicción (prólogo libro ATDD)
- Test list



## 2. Debate >> Tipos de pruebas



Ian Cooper

- Test caja negra y contrato
- BDD y Dan North



David Heinemeier

- Cucumber e infraestructura
- Quimera



## 2. Debate >> Tipos de pruebas

- Tests unitarios = visión de tunel



Bertrand Meyer

- Pirámide de tests
- ATDD visión distinta a UT (requisitos)
- Approval testing



Emily Bache

## 2. Debate >> Tipos de prueba



¿% de tipos de prueba?



# Ejemplos



## This is a vertical abstract tapestry by Robert Rauschenberg. The artwork is a dense, colorful composition of various geometric and organic forms. It features a mix of primary and secondary colors, including red, blue, yellow, green, and purple, set against a light, off-white background. The design includes concentric circles, triangles, lines, and swirling patterns, creating a complex and visually rich texture. The overall effect is one of dynamic energy and intricate detail.



- Concretar contextos
- Experimentar y extralimitarse





# Ejemplos

	Diseño previo	London / Chicago	Velocidad	Aislamiento	% ATDD
Proyecto simple	Poco	Chicago	Baja	Alto (Chicago)	Bajo
Spike	Poco	Chicago	Baja	Alto (Chicago)	Bajo
MVC CRUD	Poco	London	Alta	Bajo (Mocks)	Alto (London)
DDD + Hexagonal + microservicios	Mucho	Chicago	Baja	Alto (Arquitectura)	Medio
		London	Alta		Alto
UI React	¿?	Chicago	Baja	Alto	Bajo
		London	Alta	Bajo	Alto
Legacy	Poco / Medio	Chicago	Baja	Alto	Alto (legacy)
		London	Alta	Bajo	

# Conclusiones



# Conclusiones

- Lenguaje metafórico y confusión
- Debate vigente -> concreción
- **Con TDD no emerge el diseño de alto nivel**
- Es para gente que sabe diseñar
- Útil para spikes y requisitos claros
- Velocidad adaptable
- Aislamiento = trade off (conocimientos diseño)
- Unit tests no son suficiente (requisitos)