

ArgentinaPrograma
YoProgramo

TDD: test-driven development

por Leonardo Blautzik, Federico Gasior y Lucas Videla

Julio / Diciembre 2021



Ministerio de
Desarrollo Productivo
Argentina

¿Cómo construimos software?



Figure 1: Código primero

Dando vuelta el método



Figure 2: Prueba primero

Agregando una vuelta de tuerca

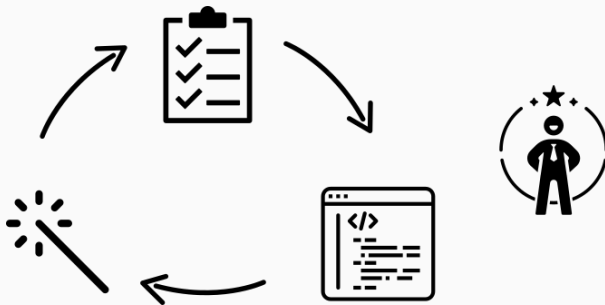


Figure 3: TDD

TDD: ¿Qué es?

Es una técnica de programación, por la cual el código se escribe para satisfacer pruebas que ponen en evidencia la necesidad de dicho código. Es por ello que se llama “desarrollo conducido por las pruebas”.

- Sólo se escribe nuevo código cuando una prueba ha fallado
- Se elimina la duplicación

Características

- Es una **forma predecible** de desarrollar software.

Características

- Es una **forma predecible** de desarrollar software.
- Permite aprender todas las lecciones que **el código tiene para enseñarnos.**

Características

- Es una **forma predecible** de desarrollar software.
- Permite aprender todas las lecciones que **el código tiene para enseñarnos.**
- Mejora los resultados al **bajar la probabilidad de fallas,**

Características

- Es una **forma predecible** de desarrollar software.
- Permite aprender todas las lecciones que **el código tiene para enseñarnos**.
- Mejora los resultados al **bajar la probabilidad de fallas**,
- Permite **que el equipo tenga confianza** en las partes,

Características

- Es una **forma predecible** de desarrollar software.
- Permite aprender todas las lecciones que **el código tiene para enseñarnos.**
- Mejora los resultados al **bajar la probabilidad de fallas,**
- Permite **que el equipo tenga confianza** en las partes,
- **Está bueno :)**

1. ROJO: Escribir una prueba **que falle**

1. ROJO: Escribir una prueba **que falle**
2. VERDE: Hacer que la prueba **pase**

1. ROJO: Escribir una prueba **que falle**
2. VERDE: Hacer que la prueba **pase**
3. AZUL: **Reducir duplicación**

1. ROJO: Escribir una prueba **que falle**
2. VERDE: Hacer que la prueba **pase**
3. AZUL: **Reducir duplicación**
4. Vuelta a empezar

El mantra de un vistazo

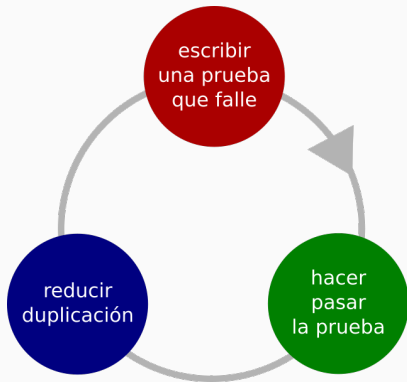


Figure 4: De un vistazo

A medida que las pruebas se vuelven más específicas, el código se vuelve más general.

- Robert Martin

Ejemplo 1, factorial

El **factorial de un entero positivo n** se define como el producto de todos los números enteros positivos desde 1 (es decir, los números naturales) hasta n

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$

$$n! = \prod_{k=1}^n k$$

Ejemplo 2, números romanos

Los **números romanos** se han utilizado por más de 3000 años. Hoy en día se utiliza el sistema arábigo para los cálculos cotidianos.

Sin embargo, los números romanos se utilizan aún para escribir números estilizados.

$$2021 = MMXXI$$

Recapitulando

- Cómo construimos software
- Invertir el proceso
- Agregar etapas y una metodología
- TDD
- Ejemplos

Para profundizar

- Beck, K. (2003). Test-driven development: By example. Addison-Wesley.
- Martin, R. The Transformation Priority Premise.
Retrieved July 28, 2021, de <https://blog.cleancoder.com/uncle-bob/2013/05/27/TheTransformationPriorityPremise.html>

¡Muchas Gracias!

continuará...



Ministerio de
Desarrollo Productivo
Argentina