



Filosofía y patrones

Qué hay que saber para programar tests.

1 Mantra

- El código de prueba no es como el código de producción.
- Simple, corto, sin abstracciones, fácil de entender.





2 Siglas y conceptos

- SUT: *System (Subject) Under Test*.
- DOCs: *Depended On Components*.

3 Secciones: Arrange, Act & Assert (AAA Pattern)

- **Arrange:** Prepara y organiza lo que necesitas.
- **Act:** Ejecuta el código y obtén una respuesta.
- **Assert:** Verifica que la respuesta es la esperada.

4 Cuestiones: Given, Should, Actual, Expected.

- **Given:**  Texto. Condiciones de la prueba.
- **Should:**  Texto. Funcionalidad esperada.
- **Actual:**  Variable. El resultado obtenido.
- **Expected:**  Variable. La respuesta esperada.

5 Test Doubles: Simuladores para no depender de las dependencias.

- **Dummy:** *(Carga previa de una base de datos)*
- **Stub:** *(Responder como lo haría un llamada http)*
- **Fake:** *(Simular una base de datos en memoria)*
- **Spy:** *(Comprobar llamadas a funciones)*
- **Mock:** *(Simular un envío de correo completo)*

6 Comprobaciones:

- igualdad
- existencia
- comparación
- pertenencia
- excepciones
- negación

7 Consejos generales

- incorpora herramientas
- evita arreglos globales
- datos realistas en los fakes
- usa etiquetas o códigos
- public black box
- evita los mocks
- haz alguna prueba