

Desarrolla y crea Test para que tus plugins no tengan errores

David Pérez



¿Qué son los Tests Automatizados?

Un test unitario es una prueba automatizada que verifica el correcto funcionamiento de la unidad más pequeña y aislada de código de un software, como una función, método o clase.



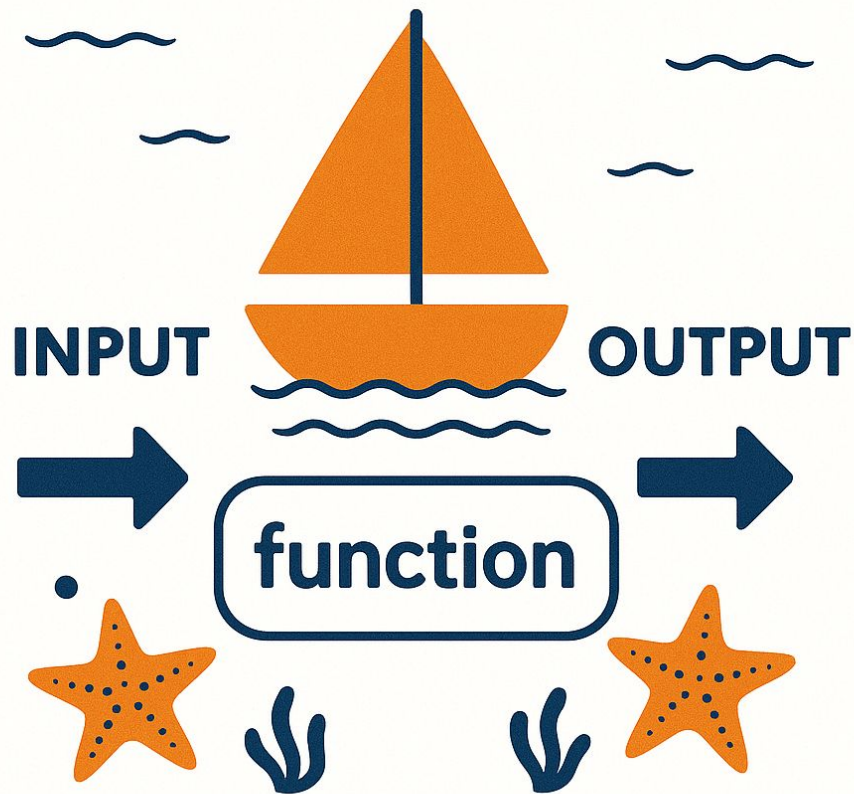
¿Qué son los Tests unitarios?

Aislamos la función

Ejecutamos la función con datos de entrada y datos de salida esperados

Desarrollamos más rápido

No hay regresiones



Ventajas

Previenen regresiones

Aumentan la fiabilidad y confianza

Facilitan el mantenimiento

Mejoran la calidad del código

Mejoran la percepción de los usuarios



Tipos de Tests

Test unitarios

Test de integración

Test funcionales

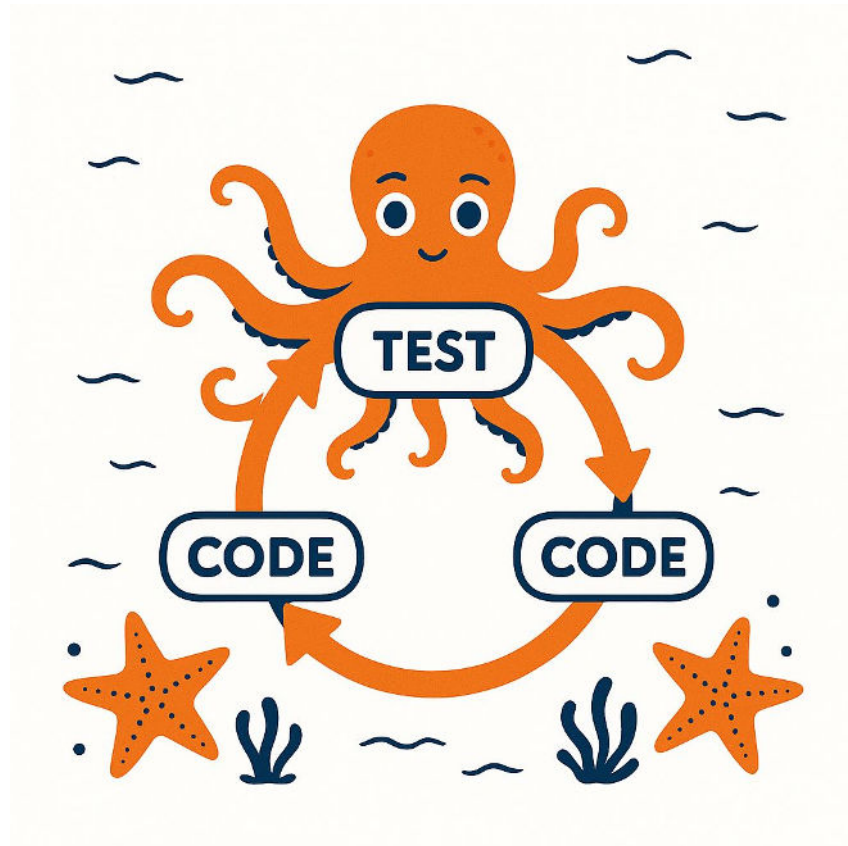


Test Driven Development

Creamos el test antes de programar

Programamos para cumplir el test

Refactorizamos hasta que se cumplen los tests

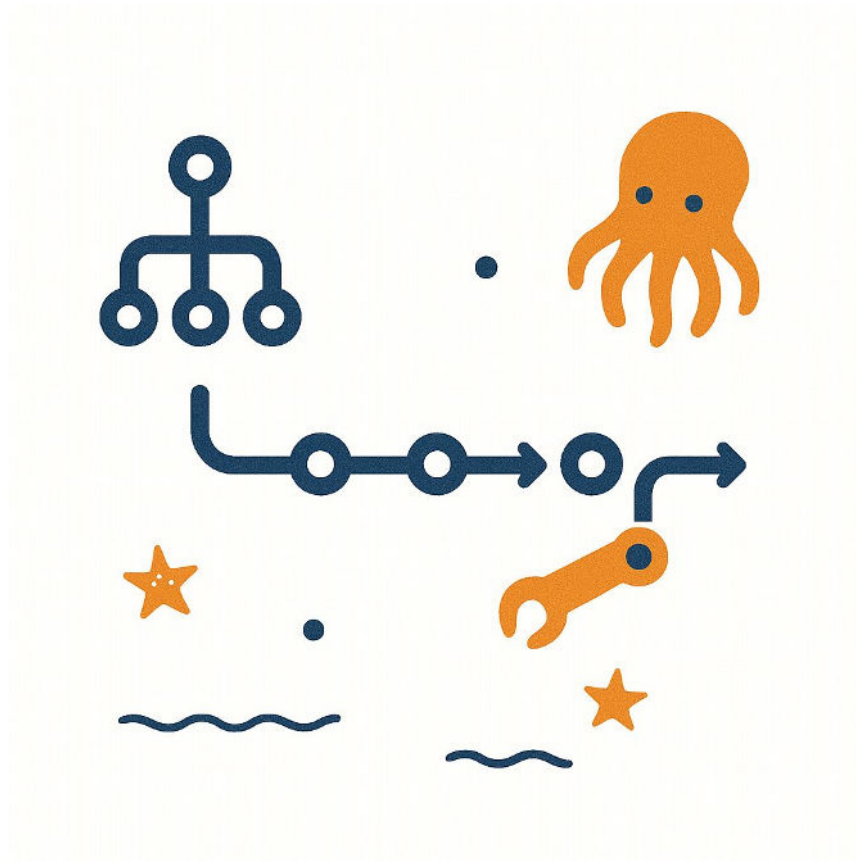


Trunk Based Development

Estrategia de Control de versiones

Rama principal donde se incorporan características nuevas

Se ejecutan Tests automatizados antes de la incorporación (Pull Request)





Checks awaiting conflict resolution

2 failing, 6 cancelled, 17 successful checks



8 failing checks ▾



PHP Code Linting / PHP (pull_request) Failing after 47s



PHP Unit Testing / PHP 7.4 - WP 6.3 (pull_request) Failing after 1m



PHP Unit Testing / PHP 7.4 - WP latest (pull_request) Cancelled after 1m



PHP Unit Testing / PHP 8.0 - WP latest (pull_request) Cancelled after 1m



PHP Unit Testing / PHP 8.1 - WP latest (pull_request) Cancelled after 1m



PHP Unit Testing / PHP 8.2 - WP latest (pull_request) Cancelled after 1m



PHP Unit Testing / PHP 8.3 - WP latest (with coverage) (pull_request) Cancelled after 1m



Afirmaciones

assertTrue(\$condition) → Checks that a condition is **true**.

assertFalse(\$condition) → Checks that a condition is **false**.

assertEquals(\$expected, \$actual)
→ Verifies that two values are equal (non-strict).

assertSame(\$expected, \$actual) → Verifies that two values are the same (strict: type and value).

assertNotEquals(\$expected, \$actual) → Verifies that two values are not equal.

assertNotSame(\$expected, \$actual)
→ Verifies that two values are not the same (strict).

assertNull(\$variable) → Checks that a variable is **null**.

assertNotNull(\$variable) → Checks that a variable is not **null**.

assertEmpty(\$variable) → Checks that a variable is empty.

assertNotEmpty(\$variable) → Checks that a variable is not empty.

assertCount(\$expectedCount, \$array) → Checks that an array (or countable) has the expected number of elements.

assertContains(\$needle, \$haystack) → Checks that a value exists within an array or string.

Afirmaciones

assertNotContains(\$needle, \$haystack) → Checks that a value does not exist within an array or string.

assertInstanceOf(\$class, \$object)
→ Verifies that an object is an instance of a given class.

assertIsArray(\$variable),
assertIsString(\$variable),
assertIsInt(\$variable), etc. → Type assertions.

assertGreaterThan(\$expected, \$actual) → Checks that a value is greater than another.

assertLessThan(\$expected, \$actual) →

Ejemplo

```
public function test_special_characters() {  
    $char_special = 'áéíóúñç';  
    $char_special_formatted = plunit_remove_special_characters( $char_special );  
    $this->assertEquals( 'aeiounc', $char_special_formatted );  
}
```



Vamos a ponerlo en Práctica

Plugin de Ejemplo:

<https://github.com/wptrainingteam/plugin-unit-tests/>

Guía:

<https://davidperezgar.com/blog/charlas/desarrolla-crea-test-plugins-no-errores/>



¡Gracias!

David Pérez

CTO Agencia CLOSE

close 

Team Rep en Plugins Team WordPress
Esponsorizado por Hostinger



davidperezgar.com/wcgalicia25

#WordCampGalicia