

ESTRUCTURA DE LAS PRUEBAS

- Las pruebas de la clase 'Coche' van dentro de una clase 'CocheTest'
- Una clase de prueba debe heredar de 'PHPUnit\Framework\TestCase'
- Los tests son públicos y empiezan por test* (o marcados con la anotación @test)
- Dentro de una prueba se encontrarán las aserciones (métodos assert) para comprobar valores reales con valores esperados

[EJEMPLO]

setUp() y tearDown()

- **setUp()**
 - Nos ayuda a configurar las pruebas y a compartir variables de configuración y objetos de prueba
 - Se ejecuta antes de cada método de prueba
- **tearDown()**
 - Se encarga de “limpiar” todo lo que se ha creado en el método setUp()
 - Se ejecuta después de cada método de prueba

[EJEMPLO]

setUpBeforeClass() y tearDownAfterClass()

- Tienen la misma funcionalidad que setUp() y tearDown(), pero se ejecutan solamente una vez:
 - setUpBeforeClass() se ejecuta antes del primer test
 - tearDownAfterclass() se ejecuta después del último test

[EJEMPLO]

DEPENDENCIAS

- Un método de prueba puede depender de otro
- PHPUnit omite ejecutar métodos si la prueba de la que depende falla
- Mejora la detección de defectos (y optimiza el tiempo de ejecución de pruebas)
- Dependencia múltiple es posible

[EJEMPLO]

PROVEEDORES

- Los métodos pueden aceptar argumentos arbitrarios (additionProvider)
- Buena práctica
- Métodos “public”
- Se ejecuta antes de setUpBeforeClass() y de setUp()

[EJEMPLO]

Pruebas Especiales - Excepciones

- La finalidad es probar (mediante el método `expectException`) si una excepción es lanzada por el código que se está probando
- Otras posibilidades:
 - `expectExceptionCode`
 - `expectExceptionMessage`
 - `expectExceptionMessageRegExp`

[EJEMPLO]

Pruebas Especiales - Errores PHP

- La finalidad es probar si un método genera la salida prevista o no
- Posibilidades:
 - `expectOutputString`
 - `expectOutputRegex`
- Utilidades:
 - `setOutputCallback`
 - `getActualOutput`

[EJEMPLO]