



PHPUnit

Pruebas Unitarias

Verificación y Validación de Software



PHPUnit



Equipo

Canul Ordoñez Josué Israel

Cimé Morales Esteban

Martín Valdez Dioné Guadalupe

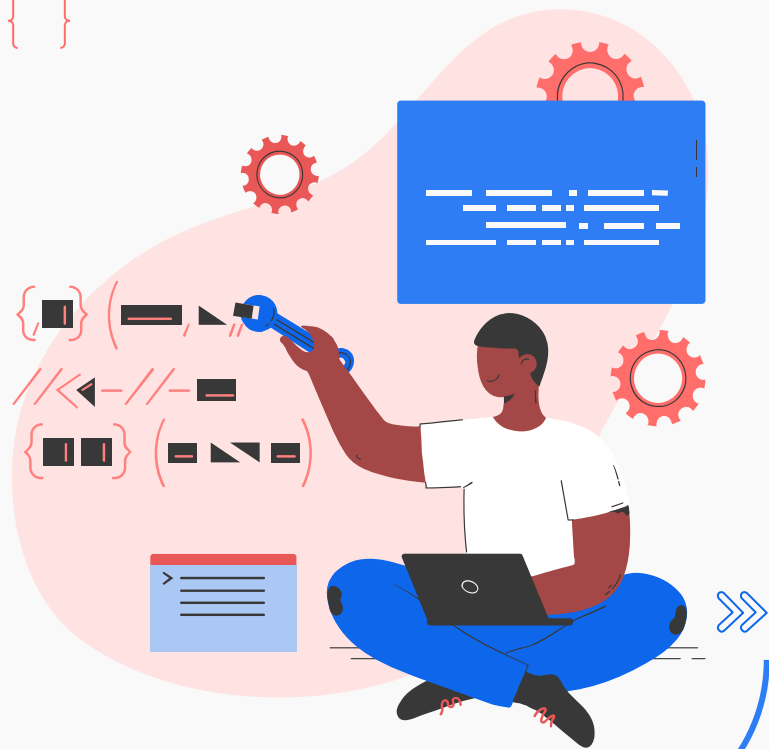
Medina Padilla Kevin Alejandro

Torres Pérez Andrea Isabel



01

Introducción



¿Qué es una prueba unitaria?

Son pruebas automatizadas que verifican de manera individual el funcionamiento de pequeñas unidades de código, como funciones o métodos.

Está diseñada para verificar que el bloque de código se ejecute según lo esperado, de acuerdo con la lógica teórica del desarrollador.

Solo interactúa con el bloque de código a través de entradas y salidas (verdaderas o falsas) capturadas afirmadas.



Características

Las pruebas unitarias **deben** de:

- Ejecutarse rápidamente
- Ejecutarse de forma independiente
- Ser deterministas, es decir, dar el mismo resultado cada vez que se ejecuten bajo mismas condiciones
- Ser automatizadas
- Ser específicas y repetibles

Ventajas

- Detección temprana de errores
 - Detección eficaz de errores
 - Mejora la mantenibilidad del código
 - Documentación del código



02

PHPUnit



{ } ¿Qué es PHPUnit?

PHPUnit es un **framework** diseñado específicamente para realizar pruebas unitarias en **aplicaciones desarrolladas en PHP**.

En términos simples, ayuda a los desarrolladores a automatizar la verificación de que cada unidad individual de código (funciones, métodos, clases) se comporte correctamente de acuerdo con lo esperado.



{ } ¿Qué es PHPUnit?

Es un **software libre** de código abierto, licenciado bajo la Licencia BSD, lo que permite su uso, modificación y distribución de manera gratuita.

Fue creado por **Sebastian Bergmann**, un desarrollador alemán reconocido en la comunidad de PHP.





Características de PHPUnit

Permite **ejecutar pruebas automáticamente** y verificar si los resultados obtenidos coinciden con los esperados.

Sigue la **estructura xUnit**, un estándar ampliamente utilizado, por lo que resulta **fácil de aprender** por su similitud con otras herramientas de test con esta estructura.



Proporciona un conjunto completo de **métodos de afirmación (assertions)** que permiten verificar todo tipo de condiciones (de igualdad hasta excepciones).





Permite crear objetos simulados (**mocks**) y objetos sustitutos (**stubs**) para aislar la unidad de código que se esté probando de sus dependencias.

Se integra fácilmente con los entornos de desarrollo integrados (IDE's), facilitando la ejecución de las pruebas, la visualización de los resultados y navegar en el código de forma eficiente.

Genera **informes detallados** sobre los resultados de las pruebas, ayudando a identificar los errores de forma rápida y rastrear la cobertura del código.



Ofrece métodos como **setUp()** y **tearDown()** para inicializar y limpiar el entorno de pruebas, asegurando que las pruebas se ejecuten en **condiciones controladas y predefinidas**.



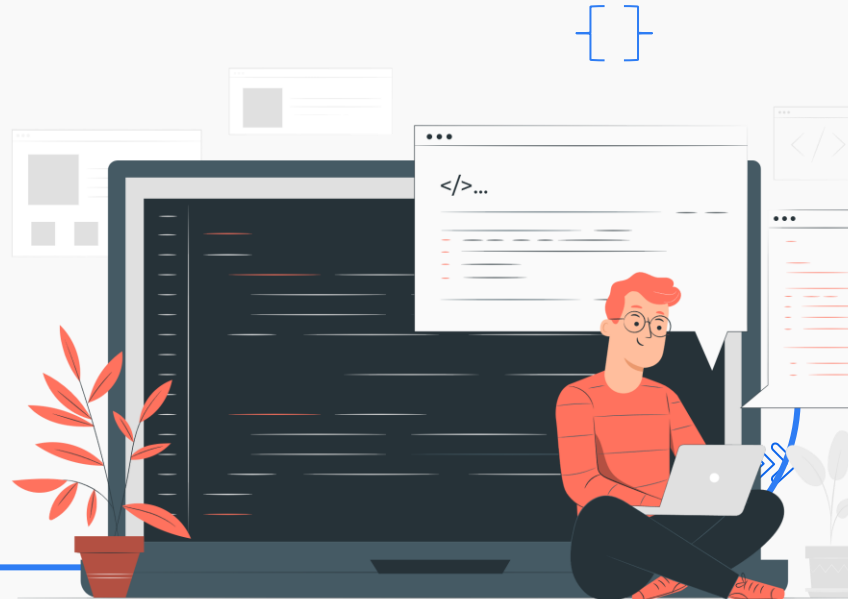
Estructura de una prueba unitaria

Una prueba unitaria en PHPUnit se define como un método dentro de una clase que extiende de `PHPUnit\Framework\TestCase`. De esta manera, este método debe verificar una funcionalidad específica del código a probar.

```
<?php

use PHPUnit\Framework\TestCase;

class MathTest extends TestCase
{
    public function testSum()
    {
        $result = 2 + 2;
        $this->assertEquals(4, $result);
    }
}
```



Aserciones (asserts)

Las aserciones son los métodos que se utilizan para verificar si el resultado de una operación es el esperado.

`assertEquals($expected, $actual)`: Verifica si dos valores son iguales.

`assertNotEquals($expected, $actual)`: Verifica si dos valores son diferentes.

`assertTrue($condition)`: Verifica si una condición es verdadera.

`assertFalse($condition)`: Verifica si una condición es falsa.

`assertNull($value)`: Verifica si un valor es nulo.

`assertNotNull($value)`: Verifica si un valor no es nulo.



Ejecución de prueba

Utilizar el comando `phpunit` en la línea de comandos desde el directorio del proyecto.



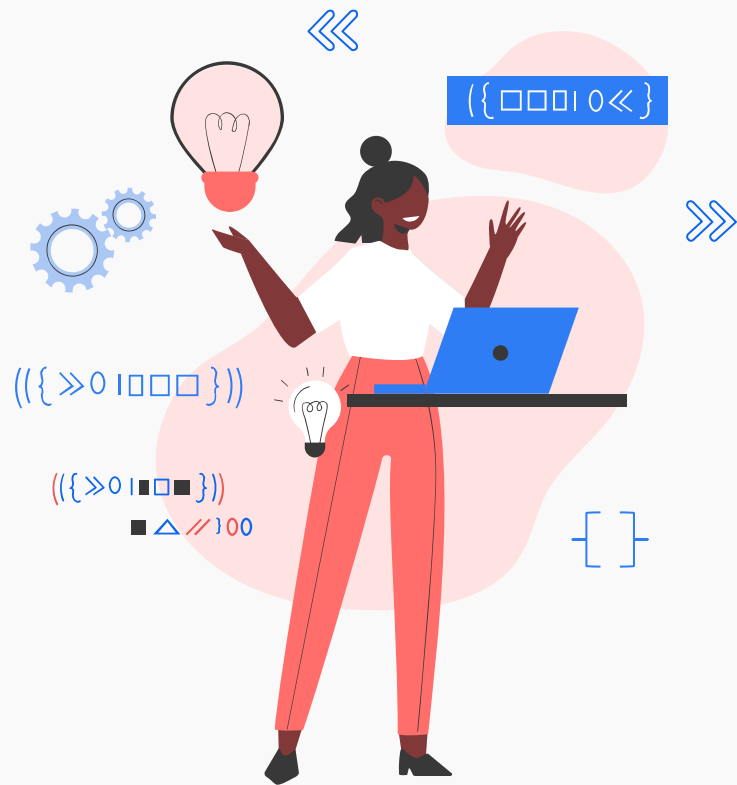
Organización de pruebas

Utilizar una clase de prueba por clase de producción. Realizar pruebas independientes.

Cobertura de código

Indica qué parte del código esta siendo ejecutada por las pruebas realizadas.





03 Sesión práctica

{ }

Requisitos de instalación

01

Composer

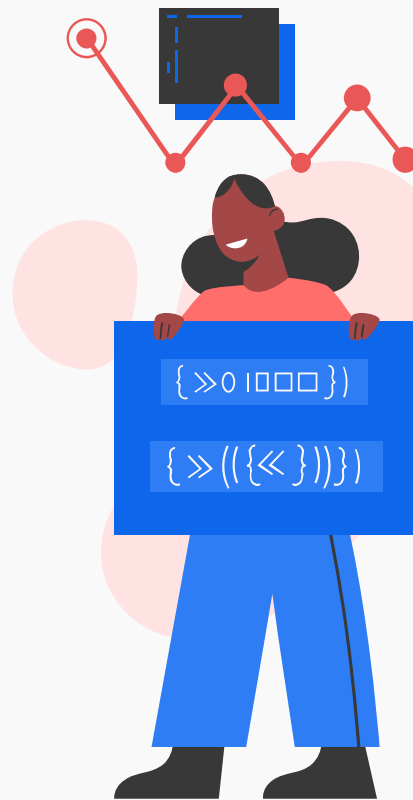
“Es una herramienta para la administración de dependencias de PHP”



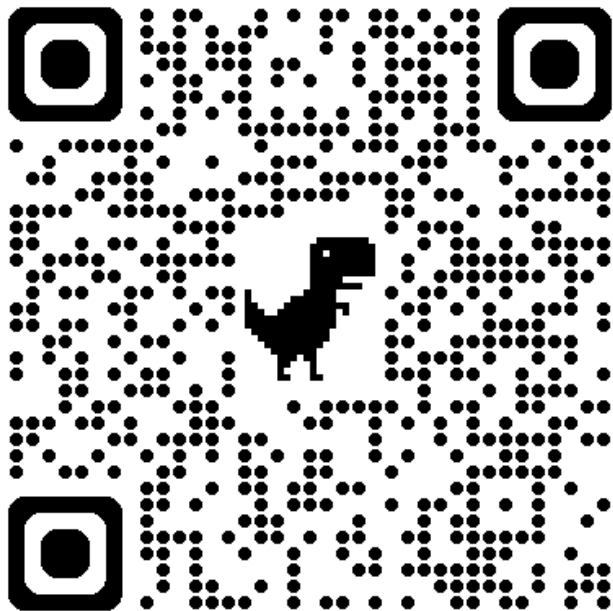
02

Php 8.2

“Es un lenguaje de programación de servidores y una potente herramienta para crear páginas web dinámicas e interactivas.”



Guía práctica



<https://github.com/KevinMed25/PHPUnit-Workshop.git>

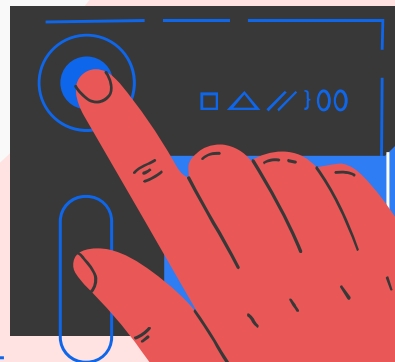
<<

[]

{ }

>>

{({({ >> }) }) << }



(({ >> 0 1 □ □ □ }))

```
((: 00 - =>> } )  
{ (<1 00 1 000 >> )}  
((: 0) > " < )  
< 01 001 } +100 0}>  
((: 0) > " < )  
{ (<1 00 1 000 >> )}
```

[]



Tarea

Instrucciones

Tomando en cuenta la clase **User.php**, realiza las pruebas unitarias en UserTest.php que se indican a continuación:

- Comprueba que la edad es correcta
- Verifica que el estado del usuario se encuentra activo (o inactivo)
- Utiliza un mock para simular el comportamiento de getName() regresando un valor diferente.



Referencias

- ¿Qué son las pruebas unitarias?: explicación de las pruebas unitarias en AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/unit-testing/>
- Mikejo. (2023, 29 noviembre). *Ejecutar pruebas unitarias mediante el Explorador de pruebas - Visual Studio (Windows)*. Microsoft Learn. <https://learn.microsoft.com/es-es/visualstudio/test/unit-test-basics?view=vs-2022>
- Recacha, R. (2020, 14 enero). Qué es PHPUnit. *OpenWebinars.net*. <https://openwebinars.net/blog/que-es-phpunit/>
- *Writing Tests for PHPUnit — PHPUnit 11.3 Manual*. (s. f.). <https://docs.phpunit.de/en/11.3/writing-tests-for-phpunit.html>
- Aweda, Z. I. (2022, 9 marzo). *How to Test PHP Code With PHPUnit*. freeCodeCamp.org. <https://www.freecodecamp.org/news/test-php-code-with-phpunit/>
- Patric. (2023, 23 febrero). A Beginner's Guide to PHPUnit: Writing and Running Unit Tests in PHP. *Medium*. <https://pguso.medium.com/a-beginners-guide-to-phpunit-writing-and-running-unit-tests-in-php-d0b23b96749f>
- I. Ali. (2022, 09 marzo). PHP Unit Testing Using PHPUnit Framework. *CloudWays*. <https://www.cloudways.com/blog/getting-started-with-unit-testing-php/>



Gracias!

Por favor, no hagan preguntas u.u