

# Testing en PHP

## Temario

1. Introducción al Testing en PHP
  - 1.1. Conceptos básicos de testing en PHP
  - 1.2. Importancia del testing en el desarrollo de software
  - 1.3. Tipos de pruebas en PHP
2. Principios Fundamentales del Testing en PHP
  - 2.1. Conceptos clave: casos de prueba, suites de pruebas, test cases, test suites
  - 2.2. Tipos de pruebas: unitarias, de integración, funcionales, de rendimiento, de seguridad
  - 2.3. Herramientas de Testing en PHP
    - 2.3.1. PHPUnit: Instalación y configuración
    - 2.3.2. PestPHP: Características, configuración y uso
    - 2.3.3. Behat: Introducción, configuración y casos de uso
3. Tests Unitarios en PHP
  - 3.1. Anatomía de un Test Unitario en PHPUnit
    - 3.1.1. Funciones de Configuración y Limpieza (SetUp y TearDown)
    - 3.1.2. Assertions: Tipos y mejores prácticas
  - 3.2. Mocking y Stubbing en PHPUnit
    - 3.2.1. Uso de PHPUnit Mock Objects
    - 3.2.2. Creación de stubs para simular comportamientos
  - 3.3. Ejercicios
4. Tests de Integración en PHP
  - 4.1. Integración de Componentes
    - 4.1.1. Pruebas de integración con Symfony y configuraciones
    - 4.1.2. Pruebas de integración con Laravel
  - 4.2. Base de Datos y Tests de Integración
    - 4.2.1. Uso de bases de datos de prueba
    - 4.2.2. Estrategias para mantener la consistencia en las pruebas
  - 4.3. Ejercicios
5. Tests Funcionales en PHP
  - 5.1. Pruebas de API
    - 5.1.1. Uso de PHPUnit para pruebas de API REST
    - 5.1.2. Estrategias para simular llamadas a servicios externos
  - 5.2. Ejercicios
6. TDD (Testing-Driven Development): método de desarrollo basado en pruebas
  - 6.1. Introducción a TDD
  - 6.2. Principios de TDD
  - 6.3. Ciclo de desarrollo en TDD
    - 6.3.1. Red, Green, Refactor
    - 6.3.2. Explicación detallada de cada fase del ciclo
  - 6.4. Implementación de TDD en PHP
    - 6.4.1. Estructura de un test en PHPUnit
    - 6.4.2. Escribir el código mínimo para pasar las pruebas
    - 6.4.3. Refactorización del código

- 6.5. Ejemplos prácticos de TDD en PHP
- 7. Cobertura de Código
  - 7.1. Uso de PHPUnit para medir la cobertura de código.
  - 7.2. Uso de Sonar
  - 7.3. Estrategias para mejorar la cobertura
- 8. Automatización de Pruebas con CI/CD
  - 8.1. Integración continua y despliegue continuo.
- 9. Buenas y malas prácticas
  - 9.1. Buenas prácticas
  - 9.2. Malas prácticas de codificación que afectan al testing
  - 9.3. Malas prácticas y antipatrones para testing