

O que são Testes Unitários?

*"Testam **pequenas unidades isoladas do código** (funções, métodos) para verificar se cada parte funciona corretamente."*

- ✓ **Verificam** se cada parte funciona corretamente
- ✓ **Detectam** erros precocemente
- ✓ **Facilitam** manutenção e refatoração

Por que Testar?



Evitar Bugs em Produção

Detectar e corrigir erros antes que cheguem aos usuários finais



Garantir Qualidade e Confiabilidade

Assegurar que o software funciona como esperado em diferentes cenários



Documentação do Comportamento

Testes servem como documentação viva do comportamento esperado



Facilitar Mudanças Seguras

Refatorar e atualizar código com confiança de que nada quebrou

Tipos de Testes - Resumo

Tipo	O que testa	Exemplo
Unitário	Função ou método isolado	<i>Função que soma dois números</i>
Integração	Comunicação entre módulos	<i>Sistema acessando banco de dados</i>
Sistema	Sistema completo	<i>Teste da funcionalidade total</i>

Estrutura de um Teste Unitário

1

Preparação (Setup)

Configurar dados e ambiente necessários para o teste



2

Execução da Função ou Método

Chamar a função/método a ser testado com os dados preparados



3

Verificação do Resultado (Assert)

Comparar o resultado obtido com o resultado esperado



Exemplo Simples em PHP - Função Soma

Uma função PHP simples que soma dois números, servindo como base para demonstração de teste unitário:

```
arquivo: soma.php

<?php

function soma($a, $b) {
    return $a + $b;
}

?>
```

Teste Unitário para Soma usando PHPUnit

```
<?php
use PHPUnit\Framework\TestCase;

class SomaTest extends TestCase {
    public function testSoma() {
        // Teste com valores positivos
        $this->assertEquals(4, soma(2, 2));

        // Teste com valores negativos
        $this->assertEquals(0, soma(-1, 1));

        // Teste com resultado maior
        $this->assertEquals(10, soma(7, 3));
    }
}
?>
```

Instalando PHPUnit



Instalar via Composer

```
$ composer require --dev phpunit/phpunit ^9
```

Instala PHPUnit como dependência de desenvolvimento do projeto



Executar Testes

```
$ ./vendor/bin/phpunit NomeDoTeste.php
```

Executa os testes unitários e exibe os resultados no terminal

Boas Práticas em Testes Unitários



Testar uma funcionalidade por teste

Cada teste deve validar apenas um comportamento específico da função



Nomear testes de forma clara

Use nomes descritivos que indicam o que está sendo testado (ex: `testSomaComNumerosPositivos`)



Cobrir casos normais e extremos

Teste valores válidos, inválidos, zero, negativos e casos limite



Manter testes independentes

Cada teste deve funcionar isoladamente, sem depender de outros testes

Exercício para a Turma



Tarefa Principal

Criar uma **função PHP** que verifica se um número é par



O que Fazer

- Escrever testes unitários usando PHPUnit para a função
- Cobrir casos: números pares, ímpares, zero e negativos
- Executar os testes e verificar os resultados



Tempo e Formato

Duração: 30-40 minutos



Trabalho em Duplas ou Individual



Objetivo

Praticar a criação de testes unitários, entender a estrutura PHPUnit e ganhar confiança na escrita de testes para suas próprias funções