



Curso PHP

Do XLSX ao CMS (aula 7)

**Eu quero a sua
melhor suíte.**





Suíte de testes

Os testes devem atuar como uma rede de segurança para alterações de código, por isso não devemos refletir a estrutura interna do código neles, teste o comportamento observável.

([Guilherme Zarelli](#))



Tipos de testes

- Unitários
- Integração
- Ponta a ponta



Teste unitário

Teste unitário é uma metodologia de testes automatizados que leva em consideração a menor unidade do software.

Ao invés de testar dentro do código fonte e executar o código manualmente para imprimir o resultado na tela, o programador cria um projeto de teste, que executa cada trecho do código de forma automática e exibe o resultado de todos os testes.

([Estevão](#))



Testes de integração

Os testes de integração determinam se as unidades de software desenvolvidas independentemente funcionam corretamente quando estão conectadas umas às outras.

O objetivo do teste de integração, como o nome sugere, é testar se muitos módulos desenvolvidos separadamente funcionam juntos conforme o esperado.

([Guilherme Zarelli](#))



Testes ponta a ponta

O teste de ponta a ponta (end-to-end) é uma metodologia de teste de software para testar um fluxo de aplicativo do início ao fim.

O objetivo deste teste é simular um cenário real do usuário e validar o sistema em teste e seus componentes para integração e integridade dos dados.

Os testes end-to-end também exigem muita manutenção e sua execução normalmente é lenta.

([Guilherme Zarelli](#))



Dublês de teste

Um dublê é um objeto falso, ou mock, construído a partir de uma classe.

Geralmente criamos esse objeto para utilizá-lo em lugar de uma dependência do cenário a ser testado.

Assim, tornamos o código de teste isento a erros indiretamente relacionados ao objetivo principal em um determinado contexto de teste.

([Estevão](#))



Como testar

Ao invés de...

Se eu digitar y, o método chamará a classe A primeiro, depois chamará a classe B e retornará o resultado da classe A mais o resultado da classe B?

Faça...

Se eu inserir os valores y, o resultado será z?

([Guilherme Zarelli](#))



O problema de testar

Testes devem ser claros, executar tarefas simples e facilitar o uso.

É necessário ficar atento para não escrever testes

- Lentos
- Acoplados
- Saibam sobre o código testado
- Não sejam confiáveis

([Diogo Winck](#))

Ferramentas de teste



PHPUnit

O PHPUnit é um framework para realização de testes unitários em PHP com base na arquitetura xUnit.

([Estevão](#))



Pest

Pest é uma estrutura de teste de unidade para PHP que oferece uma sintaxe diferente do PHPUnit.

Em particular, inspirado pelo que existe no lado JavaScript, com Jest.

No mecanismo, ele depende do PHPUnit, por isso pode ser considerado uma sobreposição do PHPUnit.

([Arthur Weill](#))



Pest: instalação

Para instalar, execute o comando:

- `composer require pestphp/pest --dev --with-all-dependencies`

Disponível em: <https://pestphp.com/docs/installation>

Após a instalação, é necessário inicializar o Pest no projeto. Para isto, execute o comando:

- `./vendor/bin/pest --init`

Escrevendo testes



O diretório tests

Por padrão, conforme definido no arquivo **phpunit.xml**, todos os testes ficam no diretório **tests**.

O arquivos devem ser nomeados terminando em **Test.php**. Por exemplo, **ExampleTest.php**.



Executando os testes

Para iniciar os teste, execute:

- **composer test**

Se observar o arquivo **composer.json**, verá que, na parte de scripts, há a referência ao comando original:

- **“test”: “./vendor/bin/pest”**

É mais fácil lembrar a forma curta do comando. Além disso, caso precise configurar o comando, basta realizar diretamente no Composer.

PASS