www.devmedia.com.br
[versão para impressão]
Link original: http://www.devmedia.com.br/articles/viewcomp.asp?comp=27550

Qualidade em desenvolvimento Web: PHP com teste unitário

Veja neste artigo uma introdução ao desenvolvimento guiado por teste em aplicações web PHP com Teste Unitário.

Afinal, é possível desenvolver aplicações web em PHP utilizando as melhores práticas de desenvolvimento guiado por teste? Como aplicar o teste unitário no desenvolvimento PHP? Este artigo apresenta uma introdução ao desenvolvimento Web PHP com teste de unidades.

Segundo o TIOB, o PHP ocupa a 6 posição entre as linguagens mais utilizadas do mundo.

Position Mar 2013	Position Mar 2012	Delta in Position	Programming Language	Ratings Mar 2013	Delta Mar 2012	Status
1	1	=	Java	18.156%	+1.05%	A
2	2	=	C	17.141%	+0.05%	A
3	5	⇈	Objective-C	10.230%	+2.49%	A
4	4	=	C++	9.115%	+1.07%	A
5	3	⇊	C#	6.597%	-1.65%	A
6	6	=	PHP	4.809%	-0.75%	A

Figura 1: Ranking de utilização das principais linguagens. Fonte: www.tiobe.com.

Ainda segundo o w3techs, o PHP é a linguagem server-side mais utilizada da Internet, dominando 78,8% de toda a internet.

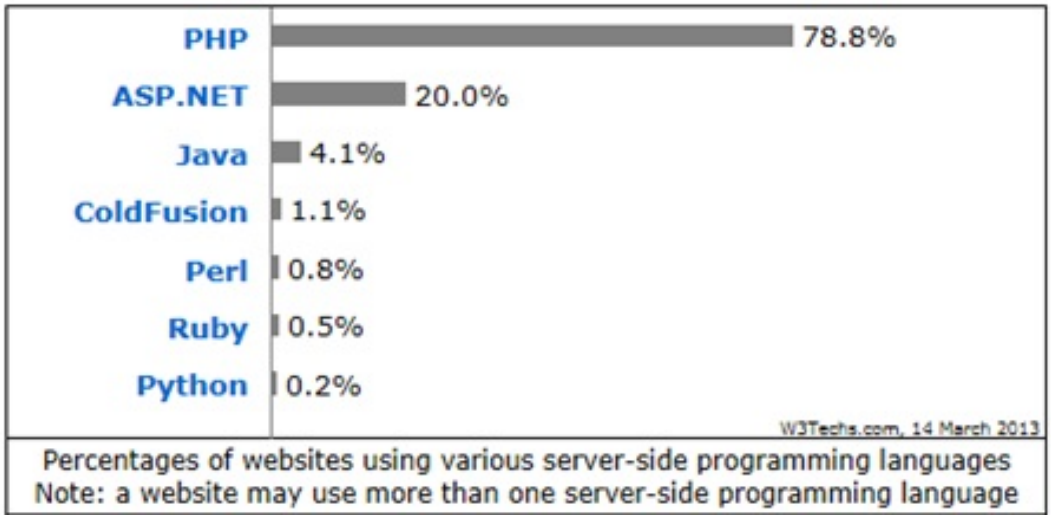

Figura 2: Ranking das principais linguagens da web. Fonte: w3techs.

Este crescimento na utilização do PHP criou uma demanda pelo desenvolvimento web com qualidade. Mas o que é qualidade?

"Qualidade de software é um processo sistemático que focaliza todas as etapas e artefatos produzidos, com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando defeitos" Alexandre Bartié.

Esta definição vai ao encontro da ideia do TDD, que tem por objetivo realizar testes de ponta a ponta, além de pensar no teste antes da implementação.

No artigo Introdução ao Desenvolvimento Guiado por Teste (TDD) com JUnit (http://www.devmedia.com.br/introducao-ao-desenvolvimento-guiado-por-teste-tdd-com-junit/26559), são apresentados conceitos importantes sobre TDD, utilizando o framwork JUnit para Java. Vamos aplicar aqui os mesmos onceitos no desenvolvimento Web PHP. Também foi tratado no artigo Qualidade de Código com PHP da importância de uma boa codificação e de ferramentas que auxiliam no processo (http://www.devmedia.com.br/qualidade-de-codigo-com-php/18119). Este artigo é também um complemento para um desenvolvimento com qualidade.

A qualidade, para ser garantida em um processo de desenvolvimento de software, deve ser acompanhada do início ao fim. Na fase inicial, o teste unitário é essencial em diversas metodologias.

Em TDD, por exemplo, o teste unitário é de extrema importância, pois é a primeira fase do processo de desenvolvimento. Veja figura a seguir, é ainda a parte central da Extreme Programming (XP), sendo recomendado ainda pela metodologia Cristal Clear e utilizado no Scrum.

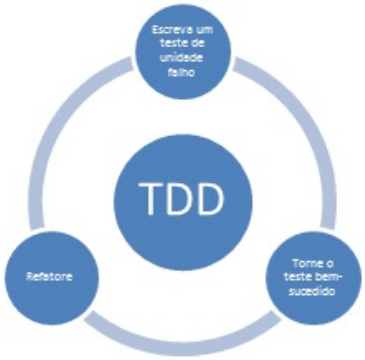

Figura 3: Ciclo de funcionamento do TDD

Segundo Leonardo Molinari, o teste de unidade é realizado em nível de componente ou classe, sendo o teste cujo o objetivo é um "pedaço do código", ou uma unidade "lógica" ou "física" do sistema. Ainda segundo Bartié, "A validação da unidade de software é a primeira etapa do processo de validação, que tem por objetivo testar os componentes individuais de uma aplicação".

Segundo a definição do RUP, o teste unitário é implementado com base no menor elemento testável (unidades) do software e implica em testar a estrutura interna (como fluxo lógico e de dados), a função da unidade e os comportamentos observáveis. O design e a implementação de testes com ênfase na estrutura interna de uma unidade se baseiam no conhecimento da implementação da unidade (abordagem caixa branca). No entanto, o design e a implementação de testes com a finalidade de verificar os comportamentos observáveis e as funções da unidade, não se baseiam no conhecimento da implementação, por isso, são conhecidos como abordagem caixa preta.

Regra de ouro do TDD: nunca escrever funcionalidade sem antes ter um teste.

Teste unitário no PHP

Como vimos, o teste unitário trabalha com base na teoria atômica, ou seja, a menor porção de um sistema, neste caso, a menor porção de um código. Aqui serão apresentadas duas formas de realizar testes unitários em PHP, vamos lá.

O PHPUnit é um framework open source de teste unitário, baseado no JUnit, com suporte a testes automatizados na linguagem PHP. Por ser baseado no JUnit, segue seu formato, tornando mais simples a criação de códigos automatizados para testes de unidade.

Um outro ponto positivo do PHPUnit é a sua integração com os principais IDE´s do mercado, como Netbeans e Eclipse, tornando o processo de teste mais fácil e simples.

Entre as vantagens do PHPUnit estão:

- Permite a criação de códigos de testes com agilidade;
- Amplamente utilizado pela comunidade de código aberto.

Para efetuar a instalação, vamos registrar o canal do PHPUnit no ambiente Pear.

Pear channel-discover pear.phpunit.de (também é possível fazer a instalação através do apt-get install phpunit no debian/ubuntu).

Agora vamos instalar o PHPUnit: Pear install phpunit/PHPUnit.

Para utilizar o PHPUnit no Netbeans, faça o seguinte:

Menu ferramentas -> opções > PHP -> teste de unidade. No campo script PHPUnit, clique em buscar e selecione o arquivo phpunit.bat. Pronto, o Netbeans está preparado para criar testes unitários e administrar os testes unitários do PHPUnit.

No exemplo abaixo, vamos fazer o processo manual, não através do IDE, para efetuar os testes unitários.

Listagem 1: Arquivo de classe que deve ser utilizado no teste - Calc.class.php

```
class CalcClass{
        public function somar($a, $b){
                return $a+$b;
        }

        public function subtrair($a, $b){
                return $a-$b;
        }
}
```

```
?>
```

Note que criamos uma classe simples com o nome CalcClass, que possui duas funções, a somar e a subtrair. Agora vamos criar nosso arquivo de teste e realizar os testes.

Listagem 2: Arquivo Calcteste.php

```
require_once 'PHPUnit/Framework.php';
//inclui o framework do phpunit
require_once diname(__FILE__)'../calc.class.php';

// indica o arquivo de classe a ser testado
class CalcClassTest extends PHPUnit_Framework_TestCase
{
        protected $object;
        protected function setup()
        {
                $this->object = new CalcClass;
        }

        public function testSomar()
        {
                $test=new CalcClass();
                $this->assertEquals('4', $test->somar(2,2));
                // verifica se é igual
        }

        public function testSubtrair()
        {
                $test=new CalcClass();
                $this->assertGreaterThan(1, $test->subtrair(10,1));
                //verifica se é maior que
        }
}
?>
```

Para executar, faça: phpunit Calcteste.php.

As asserts aqui são semelhantes às asserts do JUnit, para mais detalhes veja o artigo sobre teste com JUnit na seção Referências.

Outra forma de efetuar os testes unitários, e que eu pode ser considerada mais simples e prática, com muitos recursos, inclusive para outros testes que não sejam os unitários, é a utilização do framework simpletest, criado por Marcus Baker. Esse framework possui uma estrutura semelhante ao PHPUnit, mas com algumas mudanças para tornar mais simples. Além disso, possui uma integração direta com códigos e não há necessidade de executar através de comandos, como o PHPUnit ou de integração com o IDE.

Para começar, faça o download do SimpleTest em www.simpletest.org. Para utilizar, basta descompactar o arquivo no diretório dos arquivos que serão testados.

Continuando o exemplo acima, imagine que foi descompactado no diretório da classe, criando o diretório simpletest.

Listagem 3: Classe TestOfCalculadora

```
require_once "simpletest/autorun.php";
require_once 'calc.class.php';

class TestOfCalculadora extends UnitTestCase{
        function testSomaDoisNumerosInteiros(){
                $calculadora = new CalcClass();
                $this->asserEqual($calculadora->somar(2,"2"), 4);
                //verifica se a soma retorna 4 como igualdade
        }
}
?>
```

Para rodar o exemplo do simpletest, basta abrir o navegador, apontando para o arquivo de teste. Ele traz o teste completo com o tempo de execução e os possíveis erros.

Estes exemplos são apenas uma parte do que é possível fazer com ambas as ferramentas. O simpletest é um canivete suíço para testes em códigos PHP, possui recursos MOCK, form test, authentication test, group test, entre outros. Para mais informações sobre o simpletest, acesse o site e veja o seu manual.

Referências

- Garantia da qualidade de Software - Ed. Campus. Alexandre Bartié.
- Teste de performance - Ed. Visual Books. Leonardo Molinari.

Fabio Gomes Rocha