

Consolidando o seu conhecimento

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) Dentro da pasta **tests/Service**, crie o arquivo `EncerradorTest.php`, com o seguinte conteúdo:

```
<?php

namespace Alura\Leilao\Tests\Domain;

use Alura\Leilao\Dao\Leilao as LeilaoDao;
use Alura\Leilao\Model\Leilao;
use Alura\Leilao\Service\Encerrador;
use PHPUnit\Framework\TestCase;

class EncerradorTest extends TestCase
{
    public function testDeveEncerrarLeiloesComMaisDeUmaSemana()
    {
        $leilaoFiat = new Leilao('Fiat 147 0Km', new \DateTimeImmutable('8 days ago'));
        $leilaoVariante = new Leilao('Variante 0Km', new \DateTimeImmutable('10 days ago'));

        $leilaoDao = new LeilaoDao();
        $leilaoDao->salva($leilaoFiat);
        $leilaoDao->salva($leilaoVariante);

        $encerrador = new Encerrador();
        $encerrador->encerra();

        $leiloesEncerrados = $leilaoDao->recuperarFinalizados();
        static::assertCount(2, $leiloesEncerrados);
        static::assertEquals(
            'Fiat 147 0Km',
            $leiloesEncerrados[0]->recuperarDescricao()
        );
        static::assertEquals(
            'Variante 0Km',
            $leiloesEncerrados[1]->recuperarDescricao()
        );
    }
}
```

2) Abra o terminal (ou o Prompt de Comando, se você estiver no Windows) e acesse a pasta do projeto.

3) Execute `vendor/bin/phpunit` e verifique que os testes passam.

4) Execute novamente o mesmo comando e verifique que agora um teste está falhando.

5) Na classe `Alura\Leilao\Service\Encerrador`, faça com que o `LeilaoDao` seja injetado no construtor, obtendo o seguinte resultado:

```
<?php

namespace Alura\Leilao\Service;

use Alura\Leilao\Dao\Leilao as LeilaoDao;

class Encerrador
{
    private $dao;

    public function __construct(LeilaoDao $dao)
    {
        $this->dao = $dao;
    }

    public function encerra()
    {
        $leiloes = $this->dao->recuperarNaoFinalizados();

        foreach ($leiloes as $leilao) {
            if ($leilao->temMaisDeUmaSemana()) {
                $leilao->finaliza();
                $this->dao->atualiza($leilao);
            }
        }
    }
}
```

6) De volta ao teste EncerradorTest , crie, no mesmo arquivo, a seguinte classe:

```
class LeilaoDaoMock extends LeilaoDao
{
    private $leiloes = [];

    public function salva(Leilao $leilao): void
    {
        $this->leiloes[] = $leilao;
    }

    public function recuperarFinalizados(): array
    {
        return array_filter($this->leiloes, function (Leilao $leilao) {
            return $leilao->estaFinalizado();
        });
    }

    public function recuperarNaoFinalizados(): array
    {
        return array_filter($this->leiloes, function (Leilao $leilao) {
            return !$leilao->estaFinalizado();
        });
    }

    public function atualiza(Leilao $leilao)
    {
    }
```

```
        return;  
    }  
}
```

7) Substitua as seguintes linhas no código de teste:

- `$leilaoDao = new LeilaoDao()` por `$leilaoDao = new LeilaoDaoMock()`
 - Desta forma, você utilizará o dublê de testes
- `$encerrador = new Encerrador();` por `$encerrador = new Encerrador($leilaoDao);`
 - Assim, você passará a dependência para o `Encerrador`

8) Execute novamente os testes e garanta que estão passando. Rode-os várias vezes, para garantir que agora os testes não dependem mais do banco de dados.