

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) Adicione mais duas verificações ao teste da aula anterior:

```
...
static::assertTrue($leiloesEncerrados[0]->estaFinalizado());
static::assertTrue($leiloesEncerrados[1]->estaFinalizado());
...
```

2) Exclua a classe `LeilaoDaoMock`, criada na aula anterior.

3) Execute os testes e veja-os falhar. Entenda a mensagem de erro.

4) No teste, substitua a linha que cria um `LeilaoDao`, ou seja:

- Substitua `$leilaoDao = new LeilaoDaoMock()` por `$leilaoDao = $this->createMock(LeilaoDao::class)`

5) Execute os testes e entenda a mensagem de falha. Veja que o erro mudou.

6) Adicione o seguinte código logo após a linha que cria o `$leilaoDao`:

```
...
$leilaoDao = $this->createMock(LeilaoDao::class);
$leilaoDao->method('recuperarFinalizados')
    ->willReturn([$leilaoFiat, $leilaoVariante]);
$leilaoDao->expects(self::once())
    ->method('recuperarNaoFinalizados')
    ->willReturn([$leilaoFiat, $leilaoVariante]);
...
...
```

7) Execute os testes e veja que agora todos voltaram a passar.

8) Analise o código de testes. Veja se há algum código que pode ser removido. Analise o número de verificações que há até agora.

9) Adicione o seguinte código no seu teste, logo após toda a configuração do `$leilaoDao`:

```
...
$leilaoDao->expects($this->exactly(2)
    ->method('atualiza')
    ->withConsecutive(
        [$leilaoFiat],
        [$leilaoVariante]
    );
...
...
```

10) Execute novamente os testes. Veja que agora há uma verificação a mais sendo feita.

11) Faça a injeção de dependências na classe `Alura\Leilao\Dao\Leilao`:

```
<?php

namespace Alura\Leilao\Dao;

use Alura\Leilao\Model\Leilao as ModelLeilao;

class Leilao
{
    private $con;

    public function __construct(\PDO $con)
    {
        $this->con = $con;
    }

    ...
}
```