

08

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) Utilizando qualquer cliente de banco de dados que se conecte ao SQLite, remova todos os leilões.
- 2) Em sua classe de testes, crie um atributo estático (\$pdo) e inicialize-o no método estático `setUpBeforeClass` :

```
/** @var \PDO */
private static $pdo;

public static function setUpBeforeClass(): void
{
    self::$pdo = new \PDO('sqlite::memory:');
    self::$pdo->exec('create table leiloes (
        id INTEGER primary key,
        descricao TEXT,
        finalizado BOOL,
        dataInicio TEXT
    );');
}
```

- 3) No método `setUp`, inicie uma transação:

```
protected function setUp(): void
{
    self::$pdo->beginTransaction();
}
```

- 4) Agora, no método `tearDown`, cancele (dê `rollback`) nesta transação:

```
protected function tearDown(): void
{
    self::$pdo->rollBack();
}
```

- 5) Atualize o seu método de teste para utilizar este novo atributo. O resultado final da classe deve ser semelhante ao seguinte:

```
<?php

namespace Alura\Leilao\Tests\Integration\Dao;

use Alura\Leilao\Dao\Leilao as LeilaoDao;
use Alura\Leilao\Infra\ConnectionCreator;
use Alura\Leilao\Model\Leilao;
use PHPUnit\Framework\TestCase;

class LeilaoDaoTest extends TestCase
{
```

```
/** @var \PDO */
private static $pdo;

public static function setUpBeforeClass(): void
{
    self::$pdo = new \PDO('sqlite::memory:');
    self::$pdo->exec('create table leiloes (
        id INTEGER primary key,
        descricao TEXT,
        finalizado BOOL,
        dataInicio TEXT
    );');
}

protected function setUp(): void
{
    self::$pdo->beginTransaction();
}

public function testInsercaoEBuscaDevemFuncionar()
{
    // arrange
    $leilao = new Leilao('Variante 0Km');
    $leilaoDao = new LeilaoDao(self::$pdo);
    $leilaoDao->salva($leilao);

    // act
    $leiloes = $leilaoDao->recuperarNaoFinalizados();

    // assert
    self::assertCount(1, $leiloes);
    self::assertContainsOnlyInstancesOf(Leilao::class, $leiloes);
    self::assertSame(
        'Variante 0Km',
        $leiloes[0]->recuperarDescricao()
    );
}

protected function tearDown(): void
{
    self::$pdo->rollBack();
}
```

6) Execute os testes (`vendor/bin/phpunit`) mais de uma vez. Note que agora os testes passam, independente do número de vezes que são chamados.