1) Para instalar o **PHPUnit**, abra um terminal e navegue até o projeto. Execute:

composer require --dev phpunit/phpunit ^8

2) Para guardar as classes de testes, crie uma nova pasta e subpasta na raiz do seu projeto:

tests/Service

3) Na pasta **tests/Service**, crie uma nova classe AvaliadorTest.php, com dois métodos de teste:

<?php

namespace Alura\Leilao\Tests\Service;

use Alura\Leilao\Model\Lance;

use Alura\Leilao\Model\Leilao;

use Alura\Leilao\Model\Usuario;

use Alura\Leilao\Service\Avaliador;

use PHPUnit\Framework\TestCase;

class AvaliadorTest extends TestCase

{

public function testAvaliadorDeveEncontrarOMaiorValorDeLancesEmOrdemCrescente()

{

// Arrange - Given

$leilao = new Leilao('Fiat 147 0KM');

$maria = new Usuario('Maria');

$joao = new Usuario('João');

$leilao->recebeLance(new Lance($joao, 2000));

$leilao->recebeLance(new Lance($maria, 2500));

$leiloeiro = new Avaliador();

// Act - When

$leiloeiro->avalia($leilao);

$maiorValor = $leiloeiro->getMaiorValor();

// Assert - Then

self::assertEquals(2500, $maiorValor);

}

public function testAvaliadorDeveEncontrarOMaiorValorDeLancesEmOrdemDecrescente()

{

// Arrange - Given

$leilao = new Leilao('Fiat 147 0KM');

$maria = new Usuario('Maria');

$joao = new Usuario('João');

$leilao->recebeLance(new Lance($maria, 2500));

$leilao->recebeLance(new Lance($joao, 2000));

$leiloeiro = new Avaliador();

// Act - When

$leiloeiro->avalia($leilao);

$maiorValor = $leiloeiro->getMaiorValor();

// Assert - Then

self::assertEquals(2500, $maiorValor);

}

}

4) Para rodar o teste, execute na linha de comando:

vendor\bin\phpunit --colors tests

Repare a saída do **PHPUnit**.

5) Corrija o código da classe Avaliador, já implementando a funcionalidade de menor valor:

class Avaliador

{

private $maiorValor = -INF;

private $menorValor = INF;

public function avalia(Leilao $leilao): void

{

foreach ($leilao->getLances() as $lance) {

if ($lance->getValor() > $this->maiorValor) {

$this->maiorValor = $lance->getValor();

}

if ($lance->getValor() < $this->menorValor) {

$this->menorValor = $lance->getValor();

}

}

}

public function getMaiorValor(): float

{

return $this->maiorValor;

}

public function getMenorValor(): float

{

return $this->menorValor;

}

}

6) Adicione mais dois métodos de teste na classe AvaliadorTest, para testar o menor valor:

public function testAvaliadorDeveEncontrarOMenorValorDeLancesEmOrdemDecrescente()

{

// Arrange - Given

$leilao = new Leilao('Fiat 147 0KM');

$maria = new Usuario('Maria');

$joao = new Usuario('João');

$leilao->recebeLance(new Lance($maria, 2500));

$leilao->recebeLance(new Lance($joao, 2000));

$leiloeiro = new Avaliador();

// Act - When

$leiloeiro->avalia($leilao);

$menorValor = $leiloeiro->getMenorValor();

// Assert - Then

self::assertEquals(2000, $menorValor);

}

public function testAvaliadorDeveEncontrarOMenorValorDeLancesEmOrdemCrescente()

{

// Arrange - Given

$leilao = new Leilao('Fiat 147 0KM');

$maria = new Usuario('Maria');

$joao = new Usuario('João');

$leilao->recebeLance(new Lance($joao, 2000));

$leilao->recebeLance(new Lance($maria, 2500));

$leiloeiro = new Avaliador();

// Act - When

$leiloeiro->avalia($leilao);

$menorValor = $leiloeiro->getMenorValor();

// Assert - Then

self::assertEquals(2000, $menorValor);

}

7) Novamente execute os testes na linha de comando:

vendor\bin\phpunit --colors tests

Veja a saída do **PHPUnit**.