

Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula. Caso já tenha feito, excelente. Se ainda não, siga os passos abaixo:

1) Na linha de comando gere um novo teste:

```
php artisan make:test TemporadaTest --unit
```

2) Abra a classe de teste `TemporadaTest.php`, gerada na pasta `tests/Unit/`. Adicione os métodos `setUp`, `testBuscaPorEpisodiosAssistidos` e `testBuscaTodosOsEpisodios`:

```
<?php

namespace Tests\Unit;

use App\Episodio;
use App\Temporada;
use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class TemporadaTest extends TestCase
{
    /** @var Temporada */
    private $temporada;

    protected function setUp(): void
    {
        parent::setUp();
        $temporada = new Temporada();
        $episodio1 = new Episodio();
        $episodio1->assistido = true;
        $episodio2 = new Episodio();
        $episodio2->assistido = false;
        $episodio3 = new Episodio();
        $episodio3->assistido = true;
        $temporada->episodios->add($episodio1);
        $temporada->episodios->add($episodio2);

        $temporada->episodios->add($episodio3);

        $this->temporada = $temporada;
    }

    public function testBuscaPorEpisodiosAssistidos()
    {
        $episodiosAssistidos = $this->temporada->getEpisodiosAssistidos();

        $this->assertCount(2, $episodiosAssistidos);
        foreach ($episodiosAssistidos as $episodio) {
            $this->assertTrue($episodio->assistido);
        }
    }
}
```

```

    }
}

public function testBuscaTodosOsEpisodios()
{
    $episodios = $this->temporada->episodios;
    $this->assertCount(3, $episodios);
}
}

```

3) Na linha de comando, dentro do projeto, execute os testes:

```
vendor\bin\phpunit
```

4) Para não usar o banco de verdade nos testes, configure o ambiente de teste. Para tal, crie um novo arquivo `.env.testing`, na raiz da aplicação, com o seguinte conteúdo:

```
DB_CONNECTION=sqlite
DB_DATABASE=:memory:
```

5) Na linha de comando, gere mais dois testes:

```
php artisan make:test CriadorDeSerieTest
php artisan make:test RemovedorDeSerieTest
```

6) Abra a classe de teste `CriadorDeSerieTest.php`, gerada na pasta `tests/Feature/`, e escreva o teste para a criação da série:

```

<?php

namespace Tests\Feature;

use App\Serie;
use App\Services\CriadorDeSerie;
use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class CriadorDeSerieTest extends TestCase
{
    use RefreshDatabase;

    public function testCriarSerie()
    {
        $criadorDeSerie = new CriadorDeSerie();
        $nomeSerie = 'Nome de teste';
        $serieCriada = $criadorDeSerie->criarSerie($nomeSerie, 1, 1);

        $this->assertInstanceOf(Serie::class, $serieCriada);
        $this->assertDatabaseHas('series', ['nome' => $nomeSerie]);
        $this->assertDatabaseHas('temporadas', ['serie_id' => $serieCriada->id, 'numero' => 1]);
        $this->assertDatabaseHas('episodios', ['numero' => 1]);
    }
}

```

```
}
```

7) Abra a classe de teste `RemovedorDeSerieTest.php`, gerada na pasta `tests/Feature/`. A classe deve ficar como:

```
<?php

namespace Tests\Feature;

use App\Serie;
use App\Services\CriadorDeSerie;
use App\Services\RemovedorDeSerie;
use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class RemovedorDeSerieTest extends TestCase
{
    use RefreshDatabase;

    /** @var Serie */
    private $serie;

    protected function setUp(): void
    {
        parent::setUp();
        $criadorDeSerie = new CriadorDeSerie();
        $this->serie = $criadorDeSerie->criarSerie(
            'Nome da série',
            1,
            1
        );
    }

    public function testRemoverUmaSerie()
    {
        $this->assertDatabaseHas(
            'series',
            ['id' => $this->serie->id]
        );
        $removedorDeSerie = new RemovedorDeSerie();
        $nomeSerie = $removedorDeSerie->removerSerie($this->serie->id);
        $this->assertIsString($nomeSerie);
        $this->assertEquals('Nome da série', $this->serie->nome);
        $this->assertDatabaseMissing(
            'series',
            ['id' => $this->serie->id]
        );
    }
}
```

8) De volta à linha de comando, dentro do projeto, execute os testes:

```
vendor\bin\phpunit
```

Com a saída:

OK (4 tests, 12 assertions)