

7.2 TESTS

Añadimos, además, los métodos `setUp` y `tearDown` a la clase `TestCase` dentro de la carpeta `tests`:

```
public function setUp()
{
    parent::setUp();
    $this->createApplication();
    Artisan::call('migrate:reset');
    Artisan::call('migrate');
    Artisan::call('db:seed');
}
```

7.2 TESTS

Añadimos, además, los métodos setUp y tearDown a la clase TestCase dentro de la carpeta tests:

```
public function tearDown()  
{  
    Artisan::call('migrate:reset');  
    parent::tearDown();  
}
```

7.2 TESTS

Para ejecutar los tests, llamamos al siguiente comando:

```
$ vendor/phpunit/phpunit/phpunit
```

Aunque si lo tenéis instalado de forma global, bastaría con:

```
$ phpunit
```

7.2 TESTS

Para crear un nuevo test:

```
$ php artisan make:test PostsControllerTest
```

7.2 TESTS

Como vamos a testear el `PostController` y para varias de sus acciones se necesita estar logueado, vamos a deshabilitar el *middleware* con el fin de no tener que identificarnos en cada petición:

```
use WithoutMiddleware;
```

7.2 TESTS

Por otra parte, también tenemos que deshabilitar el token CSRF para el entorno de testing. Para ello, en el archivo `app/Http/Middleware/VerifyCsrfToken.php` añadimos la siguiente función:

```
protected function tokensMatch($request)
{
    if(env('APP_ENV') === 'testing') {
        return true;
    }
    return parent::tokensMatch($request);
}
```

7.2 TESTS

```
public function testStorePost()
{
    $data = [
        'title' => 'This is the title',
        'body' => 'the body!'
    ];
    $response = $this->call('POST', 'post/store', $data);
    $newPost = \App\Post::find(4);
    $this->assertEquals($data['title'], $newPost->title);
    $this->assertEquals(4, count(\App\Post::all()));
}
```

7.2 TESTS

```
public function testDeletePost()
{
    $postId = 1;
    $this->call('GET', 'post/delete/' . $postId);
    $this->assertEquals(2, count(\App\Post::all()));
}
```


7.2 TESTS

```
public function testDeletePost()
{
    $postId = 1;
    $this->call('GET', 'post/delete/' . $postId);
    $this->assertEquals(2, count(\App\Post::all()));
}
```

FIN