



**CEET** Centro de Electricidad Electrónica y  
Telecomunicaciones

**Ficha** 2338319

**Integrantes:**

- David Mauricio Vela
- Miguel Angel Udueña
- Miguel Angel Castillo

**Proyecto** Cronnos B.

## PRUEBAS UNITARIAS EN PHP

Para verificar la versión de PHP instalada en tu sistema utilizando la consola de Windows, puedes seguir los siguientes pasos:

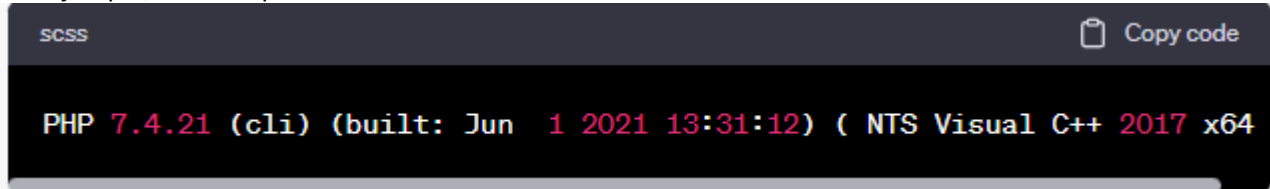
1. Abre la consola de comandos de Windows. Puedes hacerlo presionando la tecla de Windows + R, escribiendo "cmd" y luego presionando Enter.
2. En la ventana de la consola, escribe el siguiente comando y presiona Enter:

A screenshot of a Windows command prompt window. The title bar is dark gray with a 'Copy code' button on the right. The command prompt itself is black with white text. The text 'php -v' is entered at the prompt.

Este comando mostrará información sobre la versión de PHP instalada en tu sistema.

3. Verás una salida en la consola que mostrará la versión de PHP junto con otra información relevante, como el número de compilación y la configuración del entorno.

Por ejemplo, la salida podría verse así:

A screenshot of a Windows command prompt window showing the output of the 'php -v' command. The title bar is dark gray with a 'Copy code' button on the right. The command prompt is black with white text. The output is: 'PHP 7.4.21 (cli) (built: Jun 1 2021 13:31:12) ( NTS Visual C++ 2017 x64'. The text is color-coded: 'PHP' is blue, '7.4.21' is red, '(cli)' is green, '(built: Jun 1 2021 13:31:12)' is red, '( NTS Visual C++ 2017 x64' is green.

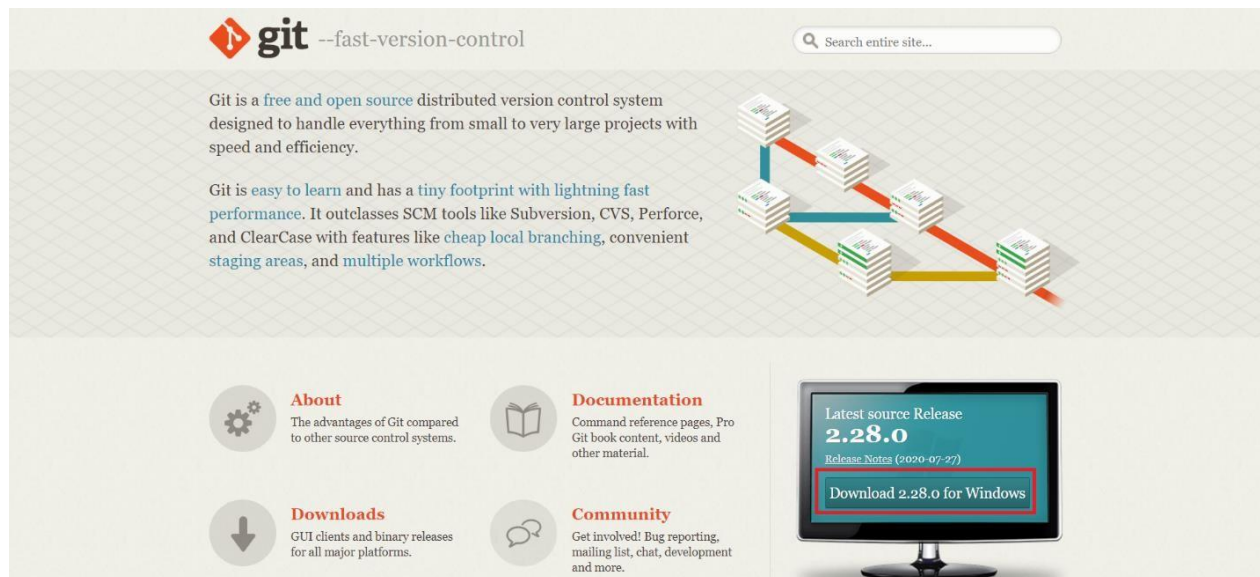
En este ejemplo, la versión de PHP instalada es la 7.4.21.

De esta manera, podrás verificar la versión de PHP instalada en tu sistema utilizando la consola de Windows.

Para descargar e instalar el programa Git, puedes seguir los siguientes pasos: Abre tu navegador web y ve a la página oficial de Git en el siguiente enlace: <https://git-scm.com/>

En la página principal, encontrarás diferentes opciones de descarga para distintos sistemas operativos. Selecciona la versión adecuada para tu sistema y haz clic en el enlace de descarga correspondiente. Una vez que se complete la descarga, busca el archivo de instalación en la carpeta de descargas de tu computadora. Haz doble clic en el archivo de instalación para iniciar el proceso de instalación de Git. Se abrirá un asistente de instalación. Sigue las instrucciones del asistente y acepta los términos y condiciones de uso. Durante el proceso de instalación, se te pedirá que elijas las opciones de configuración. Puedes dejar las opciones predeterminadas o personalizarlas según tus preferencias. Continúa con el proceso de instalación y espera a que se complete. Una vez que la instalación haya finalizado, Git estará listo para ser utilizado en tu sistema.

Para verificar que Git se ha instalado correctamente, puedes abrir la línea de comandos o la terminal y ejecutar el siguiente comando:



Configuramos GIT y por consola de Windows verificamos que haya quedado instalado:

```
C:\WINDOWS\system32\cmd.exe
Microsoft windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\>git --version
git version 2.28.0.windows.1
```

Descargar e Instalar el Programa: Composer

<https://getcomposer.org/>

[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

Download Composer Latest: v2.0.14

### Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run `Composer-Setup.exe` - it will install the latest composer version whenever it is executed.



A Dependency Manager for PHP

Latest: 1.10.13 (changelog)

A preview release for our next major version is available!  
Try out 2.0.0-RC1 (changelog) now using `composer self-update --preview`

[Getting Started](#)

[Download](#)

[Documentation](#)

[Browse Packages](#)

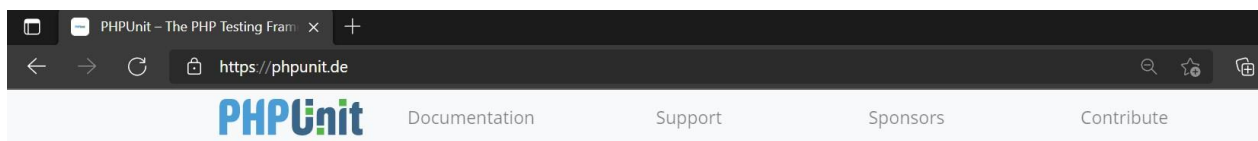
Configuramos Composer y por consola de Windows verificamos que haya quedado instalado (V mayúscula):

```
C:\WINDOWS\system32\cmd.exe  
  
C:\Users\>composer -v  
Composer version 2.0.14 2021-05-21 17:03:37
```

Es un Gestor de Paquetes de PHP

Ingresamos a la página oficial de PHPUnit:

[PHPUnit – The PHP Testing Framework](https://phpunit.de)



The currently supported versions are PHPUnit 9 and PHPUnit 8. PHPUnit 10 is currently being worked on.

Hacemos click en el botón: Take the first steps

[Take the first steps](#)

[Let me help you](#)

[Get the eBook](#)

Nos redirige a la página: **Getting Started with PHPUnit 9**

**PHPUnit 9** PHPUnit 8 PHPUnit 7 PHPUnit 6 PHPUnit 5 PHPUnit 4

## Getting Started with PHPUnit 9

This tutorial assumes that you use PHP 7.3 or PHP 7.4. You will learn how to write simple unit tests as well as how to download and run PHPUnit.

The documentation for PHPUnit 9 can be found [here](#).

### Download

#### PHP Archive (PHAR)

We distribute a [PHP Archive \(PHAR\)](#) that contains everything you need in order to use PHPUnit 9. Simply download it from [here](#) and make it executable:

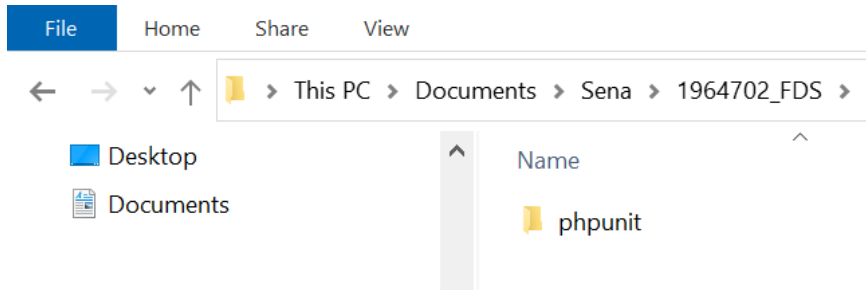
```
→ wget -O phpunit https://phar.phpunit.de/phpunit-9.phar
→ chmod +x phpunit
→ ./phpunit --version
PHPUnit 9.0.0 by Sebastian Bergmann and contributors.
```

#### Composer

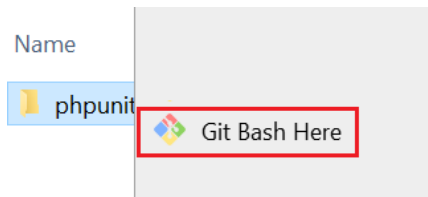
You can add PHPUnit as a local, per-project, development-time dependency to your project using [Composer](#):

```
→ composer require --dev phpunit/phpunit ^9
→ ./vendor/bin/phpunit --version
PHPUnit 9.0.0 by Sebastian Bergmann and contributors.
```

Creamos una carpeta, en este caso que se llame: phpunit

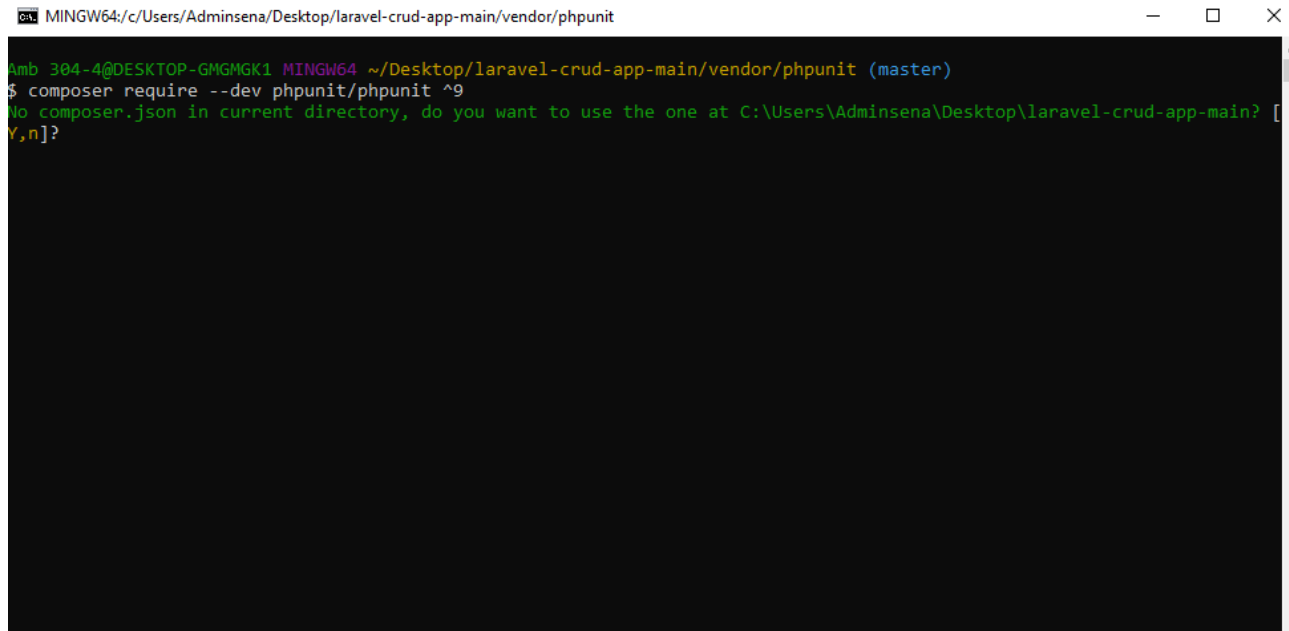


Abrimos esta carpeta usando la consola de Git:



Y allí ejecutamos el comando sugerido en la página de PHPUnit 9:

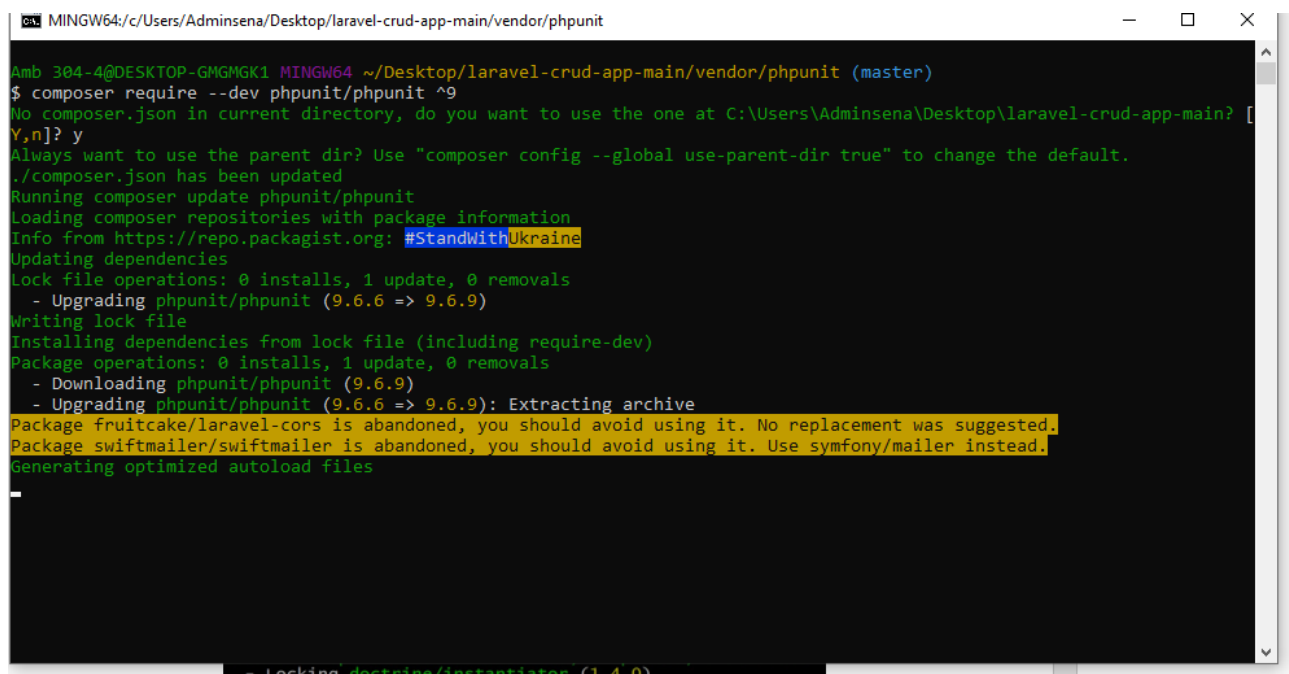
```
composer require --dev phpunit/phpunit ^9
```



```
MINGW64: c:/Users/Adminsena/Desktop/laravel-crud-app-main/vendor/phpunit

Amb 304-4@DESKTOP-GMGMGK1 MINGW64 ~/Desktop/laravel-crud-app-main/vendor/phpunit (master)
$ composer require --dev phpunit/phpunit ^9
No composer.json in current directory, do you want to use the one at C:\Users\Adminsena\Desktop\laravel-crud-app-main? [Y,n]?
```

Digitamos “Y”



```
MINGW64: c:/Users/Adminsena/Desktop/laravel-crud-app-main/vendor/phpunit

Amb 304-4@DESKTOP-GMGMGK1 MINGW64 ~/Desktop/laravel-crud-app-main/vendor/phpunit (master)
$ composer require --dev phpunit/phpunit ^9
No composer.json in current directory, do you want to use the one at C:\Users\Adminsena\Desktop\laravel-crud-app-main? [Y,n]? y
Always want to use the parent dir? Use "composer config --global use-parent-dir true" to change the default.
./composer.json has been updated
Running composer update phpunit/phpunit
Loading composer repositories with package information
Info from https://repo.packagist.org: #StandWithUkraine
Updating dependencies
Lock file operations: 0 installs, 1 update, 0 removals
- Upgrading phpunit/phpunit (9.6.6 => 9.6.9)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 0 installs, 1 update, 0 removals
- Downloading phpunit/phpunit (9.6.9)
- Upgrading phpunit/phpunit (9.6.6 => 9.6.9): Extracting archive
Package fruitcake/laravel-cors is abandoned, you should avoid using it. No replacement was suggested.
Package swiftmailer/swiftmailer is abandoned, you should avoid using it. Use symfony/mailer instead.
Generating optimized autoload files
```

Dependiendo de la velocidad que tengamos de internet, empezara la instalación de unos archivos:

Y allí ejecutamos el comando sugerido en la página de PHPUnit 9:

```
./composer.json has been created
Running composer update phpunit/phpunit
Loading composer repositories with package information
Updating dependencies
Lock file operations: 34 installs, 0 updates, 0 removals
- Locking doctrine/instantiator (1.4.0)
- Locking myclabs/deep-copy (1.10.2)
- Locking nikic/php-parser (v4.10.5)
- Locking phar-io/manifest (2.0.1)
- Locking phar-io/version (3.1.0)
- Locking phpdocumentor/reflection-common (2.2.0)
- Locking phpdocumentor/reflection-docblock (5.2.2)
- Locking phpdocumentor/type-resolver (1.4.0)
- Locking phpspec/prophecy (1.13.0)
- Locking phpunit/php-code-coverage (9.2.6)
- Locking phpunit/php-file-iterator (3.0.5)
```

Ejecutamos el comando Linux: ll (ele, ele) y observamos las carpetas que se han creado:

```
Amb 304-4@DESKTOP-GMGMGK1 MINGW64 ~/Desktop/laravel-crud-app-main/vendor/phpunit (master)
$ ll
total 24
drwxr-xr-x 1 Amb 304-4 197121 0 jun. 10 03:35 php-code-coverage/
drwxr-xr-x 1 Amb 304-4 197121 0 jun. 10 03:35 php-file-iterator/
drwxr-xr-x 1 Amb 304-4 197121 0 jun. 10 03:35 php-invoker/
drwxr-xr-x 1 Amb 304-4 197121 0 jun. 10 03:35 php-text-template/
drwxr-xr-x 1 Amb 304-4 197121 0 jun. 10 03:35 php-timer/
drwxr-xr-x 1 Amb 304-4 197121 0 jun. 18 11:06 phpunit/

Amb 304-4@DESKTOP-GMGMGK1 MINGW64 ~/Desktop/laravel-crud-app-main/vendor/phpunit (master)
```

Ejecutamos el comando Linux: `cd bin` para ingresar a la carpeta y volvemos a ejecutar el comando: `ll` (ele, ele) para observar los archivos que allí se encuentran:

```
MINGW64: c:/Users/Adminsena/Desktop/laravel-crud-app-main/tests
Amb 304-4@DESKTOP-GMGMGK1 MINGW64 ~/Desktop/laravel-crud-app-main/tests (master)
$ ll
total 17
-rw-r--r-- 1 Amb 304-4 197121 380 abr. 2 02:17 CreatesApplication.php
drwxr-xr-x 1 Amb 304-4 197121 0 abr. 2 02:17 Feature/
-rw-r--r-- 1 Amb 304-4 197121 4532 abr. 2 02:17 TestCase.php
drwxr-xr-x 1 Amb 304-4 197121 0 abr. 2 02:17 Unit/
Amb 304-4@DESKTOP-GMGMGK1 MINGW64 ~/Desktop/laravel-crud-app-main/tests (master)
$
```

Ahora ejecutamos el comando: `./phpunit` y observamos la ayuda que ofrece el comando ejecutado:

```
$ ./phpunit
PHPUnit 9.5.4 by Sebastian Bergmann and contributors.

Usage:
  phpunit [options] UnitTest.php
  phpunit [options] <directory>

Code Coverage Options:
  --coverage-clover <file>   Generate code coverage report in Clover XML
                              format
  --coverage-cobertura <file> Generate code coverage report in Cobertura XML
                              format
  --coverage-crap4j <file>   Generate code coverage report in Crap4J XML
                              format
  --coverage-html <dir>      Generate code coverage report in HTML format
  --coverage-php <file>      Export PHP_CodeCoverage object to file
```

ejecutamos el comando: `./phpunit --version` y verificamos la versión que hemos instalado, el autor y los contribuidores.

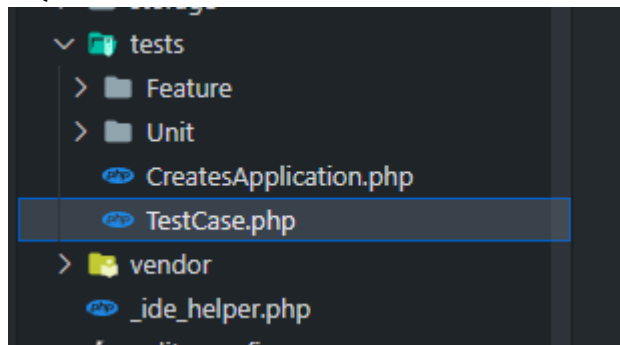
```
$ ./phpunit --version
PHPUnit 9.5.4 by Sebastian Bergmann and contributors.
```



Creamos una carpeta tests y dentro de ella un archivo: TestCase.php, allí copiamos el siguiente código:

```
1 <?php
2
3 namespace Tests;
4
5 use App\Models\Role;
6 use App\Models\RoleType;
7 use App\Models\Training;
8 use App\Models\Type;
9 use App\Models\User;
10 use Illuminate\Foundation\Testing\TestCase as BaseTestCase;
11 use Illuminate\Http\UploadedFile;
12 use Illuminate\Support\Collection;
13 use ReflectionClass;
14 use ReflectionException;
15 use Storage;
16
17 abstract class TestCase extends BaseTestCase
18 {
19     use CreatesApplication;
20
21     /** @var bool $initialised */
22     public static bool $initialised = false;
23
24     /**
25      * Setup the test environment.
26      *
27      * @return void
28      */
29     protected function setUp(): void
30     {
31         parent::setUp();
32
33         if (!static::$initialised) {
34             $this->seedRequiredData();
35
36             static::$initialised = true;
37         }
38     }
39
40     /**
41      * Creates a new Administrator
42      *
43      */
44 }
```

Quedando así:



Para ejecutar la prueba ejecutamos el comando:

```
./vendor/bin/phpunit --bootstrap ./vendor/autoload.php ./tests/PrimeraPrueba.php --colors always
```

Y revisamos el resultado obtenido:

Creamos una carpeta tests y dentro de ella un archivo: TestCase.php, allí copiamos el siguiente

```
$ ./vendor/bin/phpunit --bootstrap ./vendor/autoload.php ./TestCase.php
Colors always
PHPUnit 9.5.4 by Sebastian Bergmann and contributors.

R                                                                    1 / 1

Time: 00:00.011, Memory: 4.00 MB

There was 1 risky test:

1) PrimeraPrueba::testPrimeraPrueba
This test did not perform any assertions

[redacted]\Documents\Sena\1964702_FDS\phpunit\tests\TestCase.php

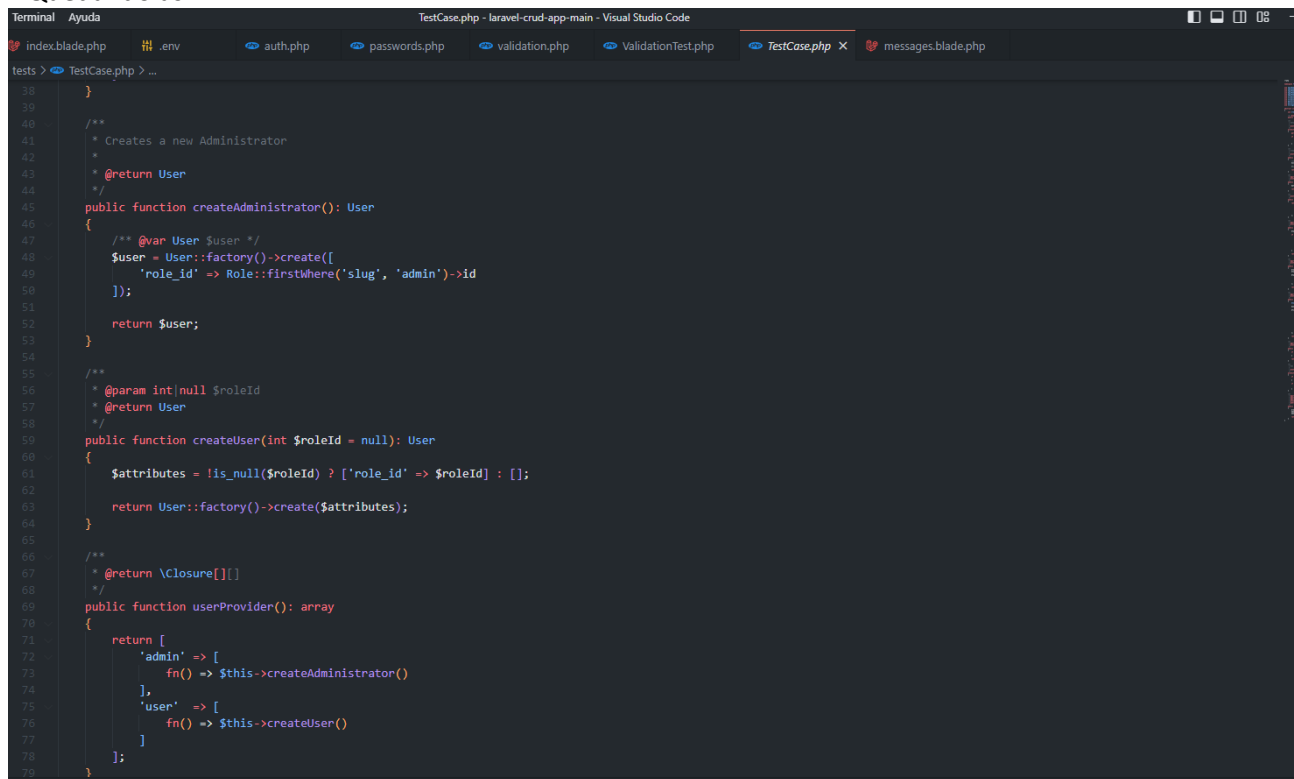
OK, but incomplete, skipped, or risky tests!
Tests: 1, Assertions: 0, Risky: 1.
```

Agregamos un assertion negativo al método:

```
<?php declare(strict_types=1);
use PHPUnit\Framework\TestCase;

final class PrimeraPrueba extends TestCase{
    public function testPrimeraPrueba(): void
    {
        $this->assertTrue(false);
    }
}
?>
```

Quedando así:



```
38     }
39
40     /**
41      * Creates a new Administrator
42      *
43      * @return User
44      */
45     public function createAdministrator(): User
46     {
47         /** @var User $user */
48         $user = User::factory()->create([
49             'role_id' => Role::firstWhere('slug', 'admin')->id
50         ]);
51
52         return $user;
53     }
54
55     /**
56      * @param int|null $roleId
57      * @return User
58      */
59     public function createUser(int $roleId = null): User
60     {
61         $attributes = !is_null($roleId) ? ['role_id' => $roleId] : [];
62
63         return User::factory()->create($attributes);
64     }
65
66     /**
67      * @return \Closure[]
68      */
69     public function userProvider(): array
70     {
71         return [
72             'admin' => [
73                 fn() => $this->createAdministrator()
74             ],
75             'user' => [
76                 fn() => $this->createUser()
77             ]
78         ];
79     }
```

Y Ejecutamos de nuevo la prueba:

```
./vendor/bin/phpunit --bootstrap ./vendor/autoload.php ./tests/TestCase.php --colors always
```

Para que quede de la siguiente manera

Agregamos un assertion negativo al método:

```
$ ./vendor/bin/phpunit --bootstrap ./vendor/autoload.php ./tests
PHPUnit 9.5.4 by Sebastian Bergmann and contributors.

F

Time: 00:00.012, Memory: 4.00 MB

There was 1 failure:

1) PrimeraPrueba::testPrimeraPrueba
Failed asserting that false is true.

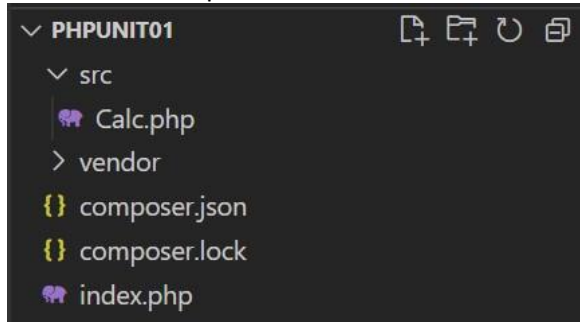
\Documents\Sena\1964702_FDS\phpunit\tests\P

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

Creamos un Proyecto: `phpunit01`

Instalamos PHPUnit: `composer require --dev phpunit/phpunit ^9`

Creamos una carpeta: `src`



En ella creamos una clase: `Calc.php`

```
<?php
class Calc{
    public function suma($a, $b){
        return $a + $b;
    }
}
?>
```

Creamos un archivo: `index.php`

```
<?php

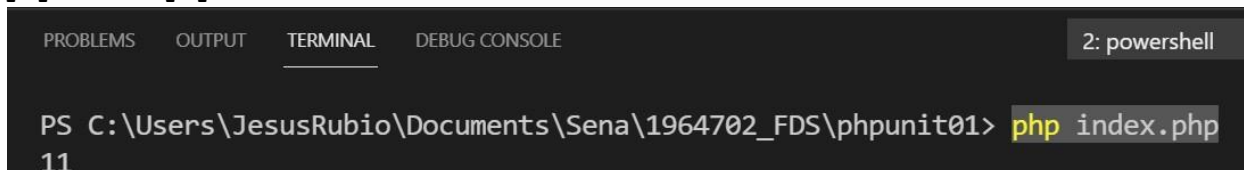
include 'src/Calc.php';

$calcInstance = new Calc();

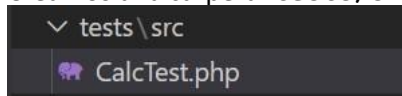
echo $calcInstance->suma(3,8);
?>
```

Probamos que este funcionando el método, ejecutando la siguiente instrucción en la terminal:

`php index.php`



Creamos una carpeta: `tests/src`



Creamos un archivo: `CalcTest.php`

```
<?php declare(strict_types=1);
include 'src/Calc.php';
use PHPUnit\Framework\TestCase;

class CalcTest extends TestCase
{
    public function testSuma()
    {
        // init
        $calcInstance = new Calc();
        $mockA = 2;
        $mockB = 3;
        $expected = 5;

        // act
        $result = $calcInstance->suma($mockA, $mockB);

        // assert
        $this->assertEquals($expected, $result);
    }
}
?>
```

Ejecutamos la prueba con la instrucción:

```
./vendor/bin/phpunit --bootstrap ./vendor/autoload.php
./tests/src/CalcTest.php --colors always
```

```
$ ./vendor/bin/phpunit --bootstrap ./vendor/autoload.php ./tests/src/CalcTest.php --colors always
PHPUnit 9.5.4 by Sebastian Bergmann and contributors.

.                                                                 1 / 1 (100%)

Time: 00:00.012, Memory: 4.00 MB

OK (1 test, 1 assertion)
```

O con la instrucción:

```
./vendor/bin/phpunit tests
```

```
$ ./vendor/bin/phpunit tests
PHPUnit 9.5.4 by Sebastian Bergmann and contributors.

.                                                                    1 / 1 (100%)
Time: 00:00.012, Memory: 4.00 MB

OK (1 test, 1 assertion)
```

O con la instrucción:

```
./vendor/bin/phpunit --testdox tests
```

```
$ ./vendor/bin/phpunit --testdox tests
PHPUnit 9.5.4 by Sebastian Bergmann and contributors.

Calc
 ✓ Suma
Time: 00:00.013, Memory: 6.00 MB

OK (1 test, 1 assertion)
```