



# UNIVERSIDAD DE COLIMA

Facultad de Telemática  
Ingeniería en Software

Práctica.- Pruebas unitarias con PHPUnit.

Desarrollo de Software Seguro

Integrantes del equipo:

Bañuelos Munguía Roberto Antonio

Ochoa Lopez Hugo Alejandro

Rodríguez Reyes Miguel Jordán

Sanchez Medina Julian

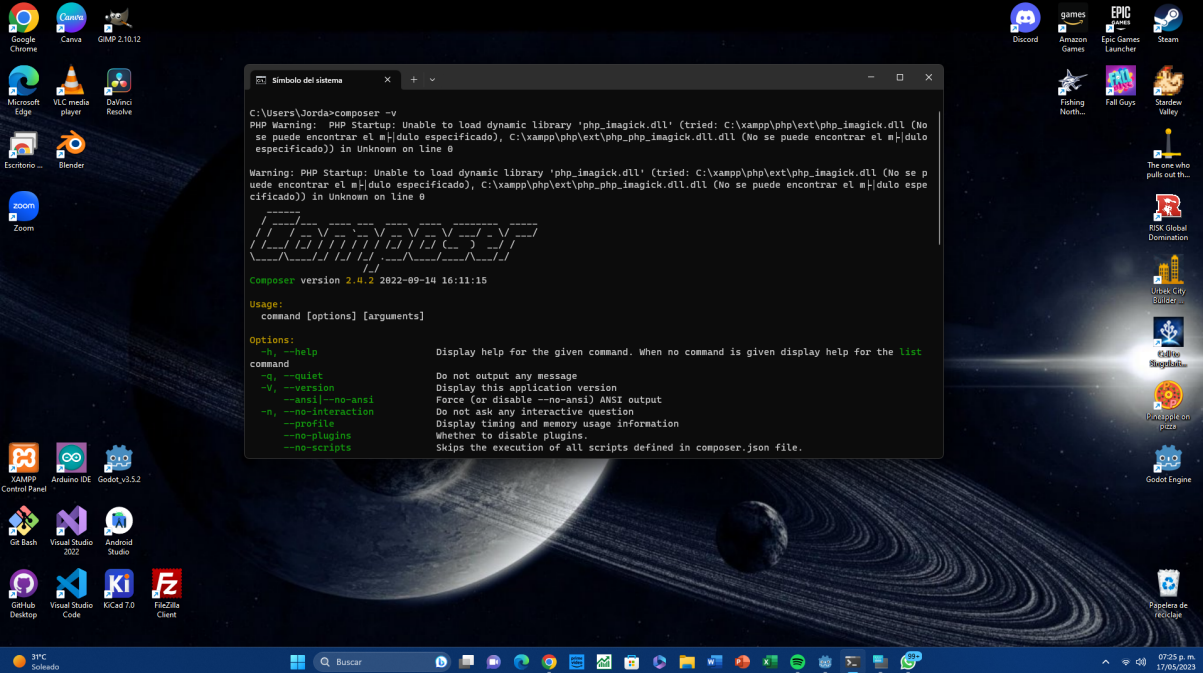
6°E

18/05/2023.

# Pruebas unitarias usando PHPUnit

En esta práctica realizaremos pruebas unitarias usando el entorno PHPUnit, para este práctica se usó como base el código fuente de una clase llamada “animal”, dicho código se encuentra en el repositorio de GitHub siguiente: <https://github.com/josena/testphpunit.git>, y con ayuda de un video de apoyo se realizaron los siguientes paso:

Primer paso es instalar composer, en nuestro caso ya contábamos con él, para verificar usamos el comando “**composer -v**”



The screenshot shows a Windows 10 desktop with a dark theme. A terminal window titled 'Símbolo del sistema' is open, displaying the output of the command 'composer -v'. The output shows a warning about the 'php\_imagick.dll' library not being found, followed by the Composer version '2.4.2' and the date '2022-09-14 16:11:15'. Below this, the 'Usage' and 'Options' are listed. The desktop background features a space-themed image with a planet and rings. Various application icons are visible on the taskbar and desktop, including Google Chrome, VLC media player, and several games.

```
C:\Users\Jorda>composer -v
PHP Warning:  PHP Startup: Unable to load dynamic library 'php_imagick.dll' (tried: C:\xampp\php\ext\php_imagick.dll (No se puede encontrar el módulo especificado), C:\xampp\php\ext\php_php_imagick.dll (No se puede encontrar el módulo especificado)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'php_imagick.dll' (tried: C:\xampp\php\ext\php_imagick.dll (No se puede encontrar el módulo especificado), C:\xampp\php\ext\php_php_imagick.dll (No se puede encontrar el módulo especificado)) in Unknown on line 0

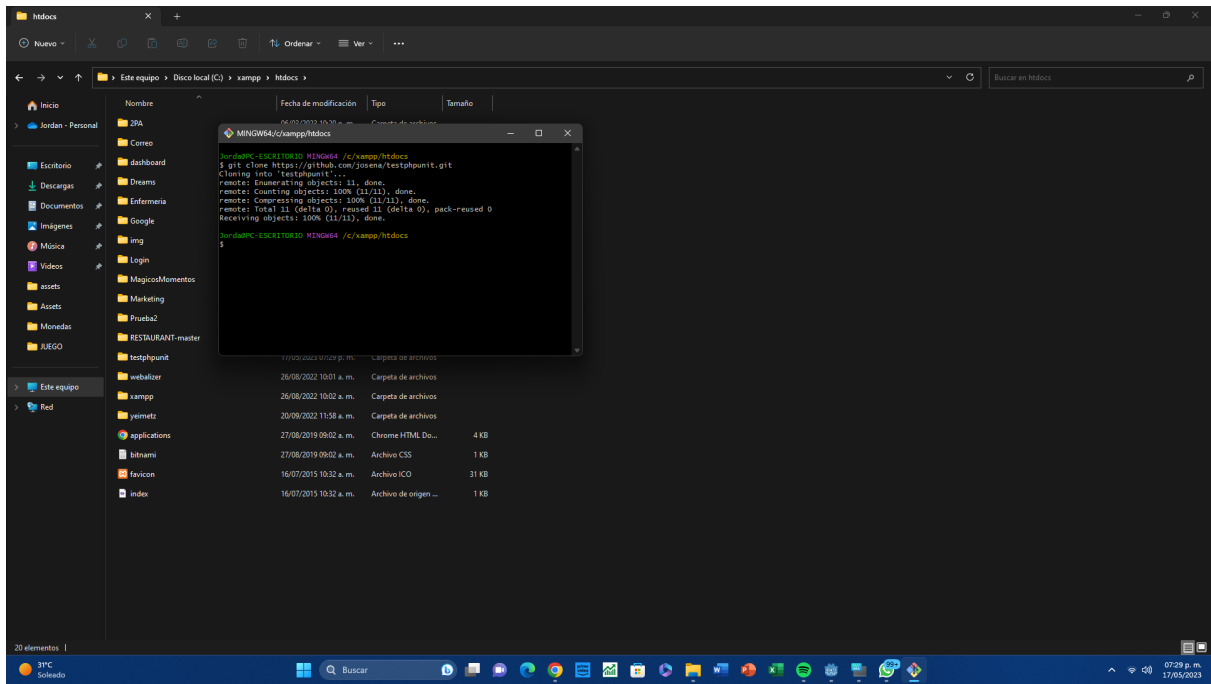
Composer version 2.4.2 2022-09-14 16:11:15

Usage:
  command [options] [arguments]

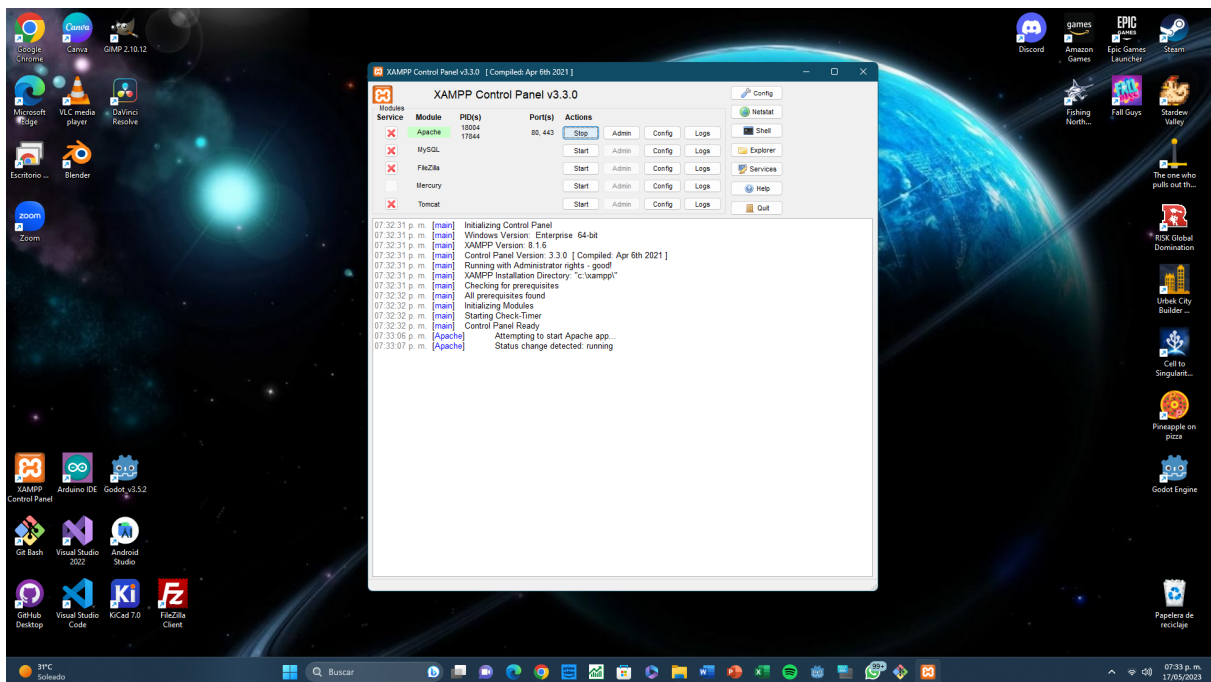
Options:
  -q, --quiet                Do not output any message
  -V, --version              Display this application version
  --ansi|--no-ansi           Force (or disable) ANSI output
  -n, --no-interaction       Do not ask any interactive question
  --profile                 Display timing and memory usage information
  --no-plugins               Whether to disable plugins.
  --no-scripts               Skips the execution of all scripts defined in composer.json file.
```

De igual manera debemos tener instalado XAMPP, en nuestro caso ya contamos con él por lo que omitiremos este paso.

Posteriormente debemos clonar el repositorio <https://github.com/josena/testphpunit.git>, para ello, como contamos con GIT instalado, utilizamos el comando `git clone https://github.com/josena/testphpunit.git`



Se debe considerar que para que pueda correr debe estar en la carpeta htdocs dentro de XAMPP. Posteriormente iniciamos el servidor APACHE en XAMPP.



Ahora ya podemos visualizar el proyecto entrando a la URL: <http://localhost/testphpunit/>

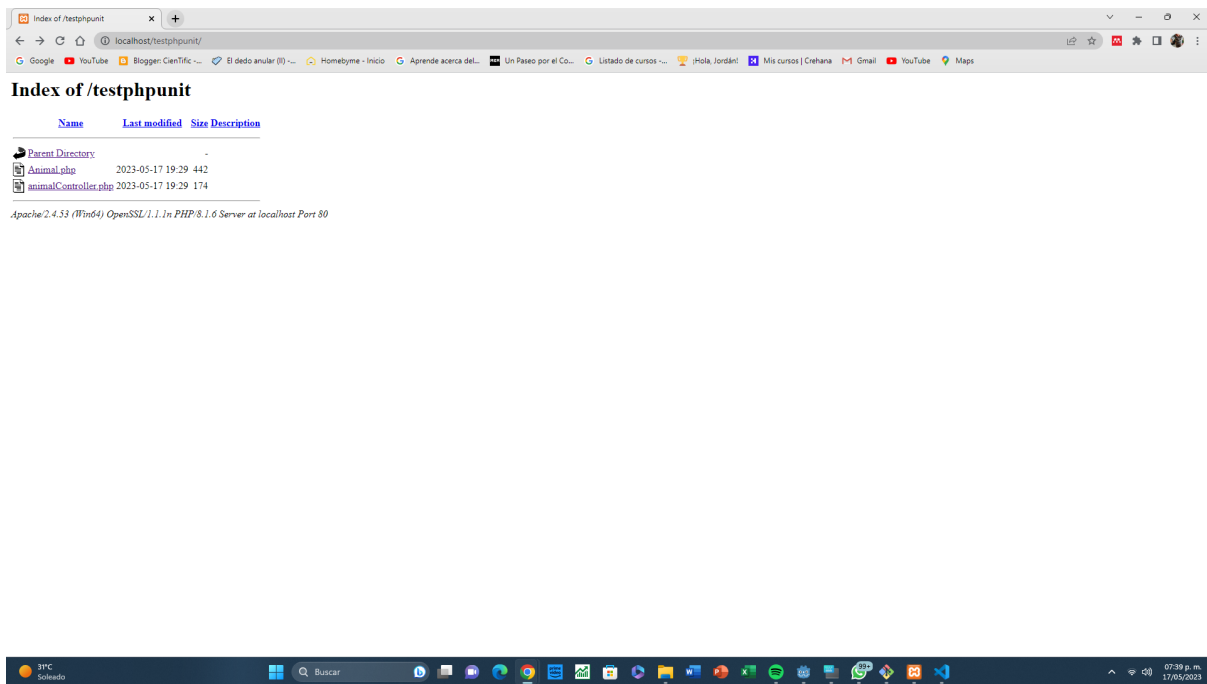
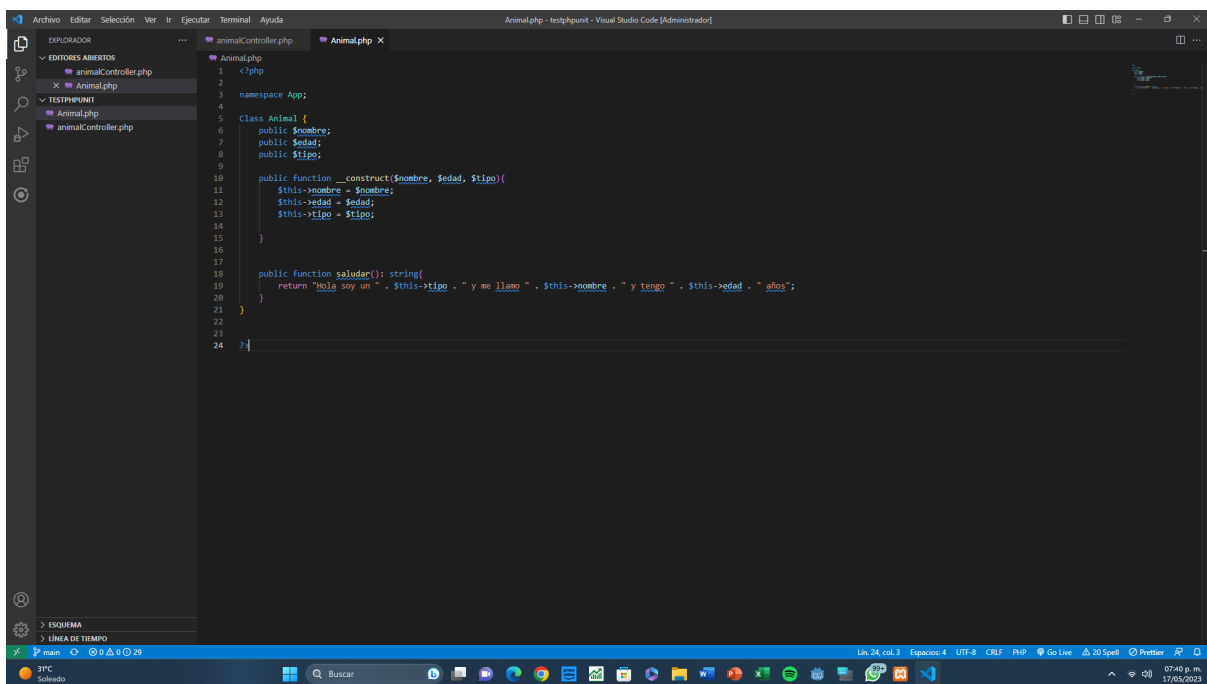


Figura 1: **Servidor APACHE en XAMPP**

Además abrimos el código en el editor de texto VS CODE.



Posteriormente tenemos que instalar **PHPUNIT**, para ello en la terminal del editor de texto ponemos el comando **“composer phpunit/phpunit ^10”**

Para saber la versión de php que tienes puedes ver en la figura 1. En la parte final aparece la versión de Apache y de PHP.

Además en base al siguiente cuadro sabrás qué versión necesitas instalar.

Major Version	PHP Compatibility	Initial Release	End of Bugfix Support	End of Life
PHPUnit 10	>= PHP 8.1	February 3, 2023	February 7, 2025	To be determined
PHPUnit 9	>= PHP 7.3	February 7, 2020	February 2, 2024	To be determined
PHPUnit 8	>= PHP 7.2	February 1, 2019	February 3, 2023	To be determined
PHPUnit 7	PHP 7.1 - PHP 7.3	February 2, 2018	February 7, 2020	February 7, 2020
PHPUnit 6	PHP 7.0 - PHP 7.2	February 3, 2017	February 1, 2019	February 1, 2019
PHPUnit 5	PHP 5.6 - PHP 7.1	October 2, 2015	February 2, 2018	February 2, 2018
PHPUnit 4	PHP 5.3 - PHP 5.6	March 7, 2014	February 3, 2017	February 3, 2017

**Current stable**  
**Previous stable**  
**End-of-Life**

**Bugfix Support** means that bugs will be fixed for a version of PHPUnit. Bugfix Support for major version  $x$  ends two years after its initial release (when major version  $(x+2)$  is released). Bugfix Support for minor version  $x.y$  ends when minor version  $x.(y+1)$  is released.

When Bugfix Support ends for a major version, Life Support for this major version begins.

**Life Support** means that changes will be made for an otherwise unsupported version of PHPUnit to be compatible with new versions of PHP. Life Support ends when an old version of PHPUnit cannot be made compatible with a new version of PHP without breaking backwards compatibility.

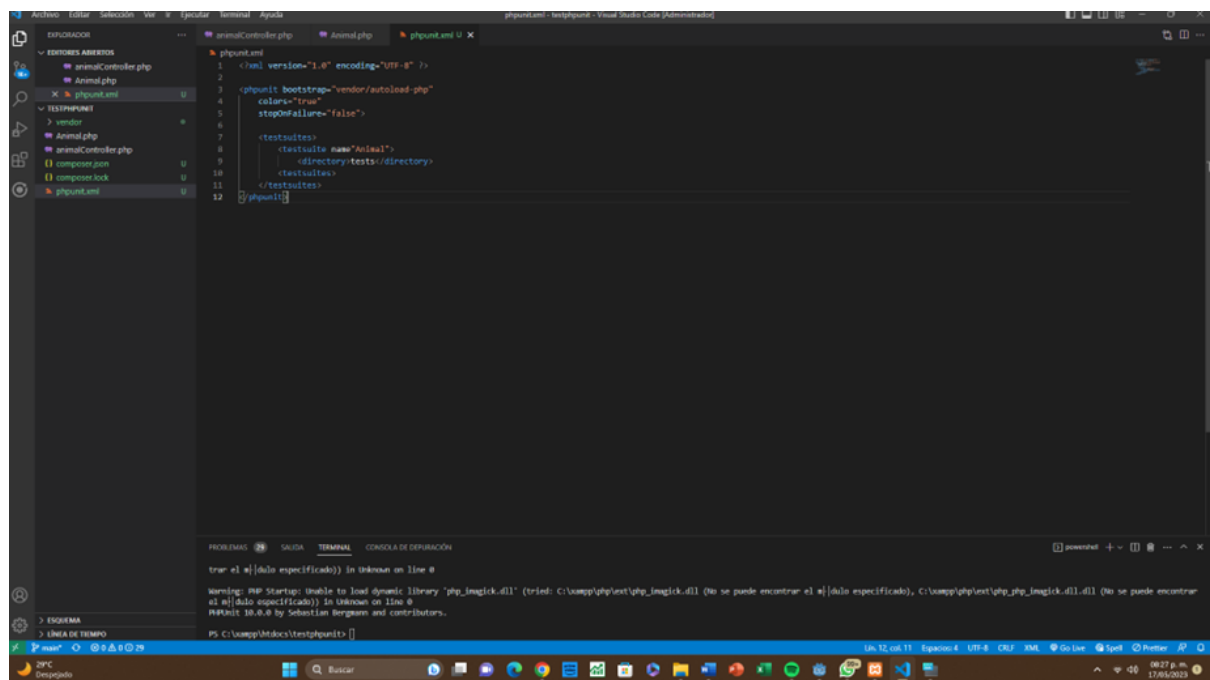
The fact that a version of PHPUnit supports a specific PHP version means that this version of PHPUnit works on that PHP version for PHP code that is compatible with the version of PHP required by the PHPUnit version in question. For example, PHPUnit 8.5 requires PHP 7.2. PHPUnit 8.5 works for PHP code compatible with PHP 7.2 on PHP 8. However, PHPUnit 8.5's test double code generator does not support PHP 8's union types.

Después de instalar **PHPUnit** podemos verificar que sea la versión correcta con el comando “**vendor/bin/phpunit --version**”, en este caso confirmamos que tenemos la versión 10.

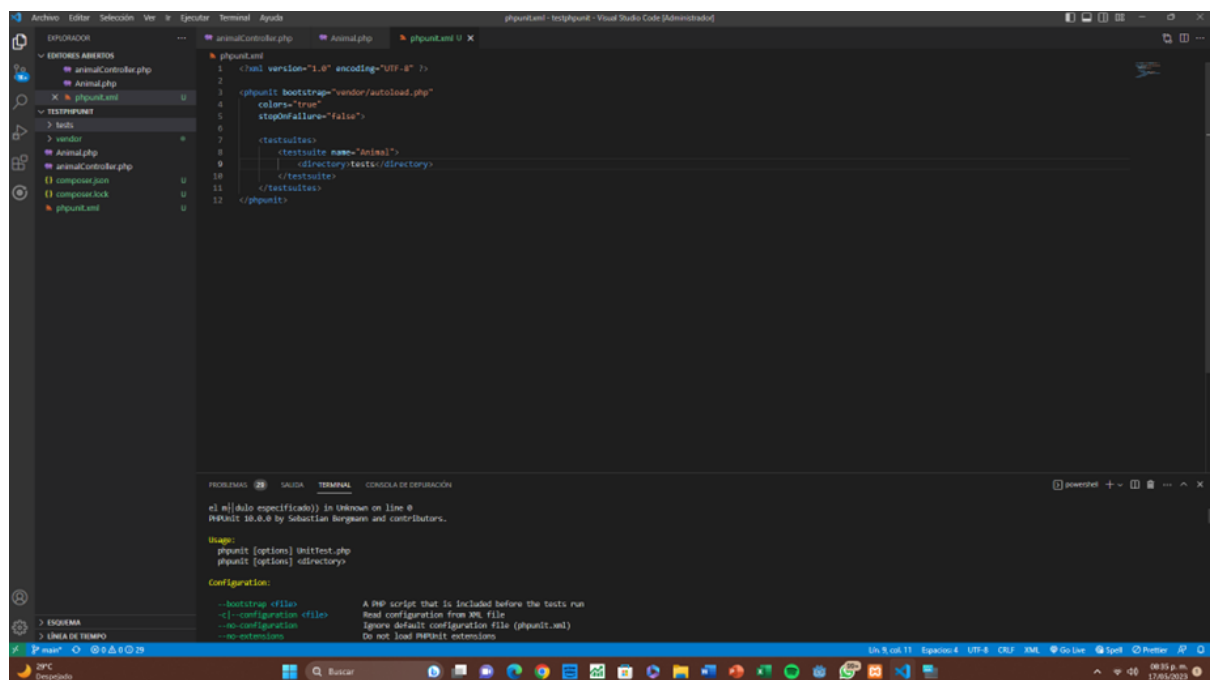
```
traz-ei m[dulo especificado]) in unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'php_imagick.dll' (tried: C:\xampp\php\ext\php_imagick.dll (No se puede encontrar el m[dulo especificado), C:\xampp\php\ext\php_imagick.dll.dll (No se puede encontrar el m[dulo especificado)) in unknown on line 0
PHPUnit 10.0.0 by Sebastian Bergmann and contributors.

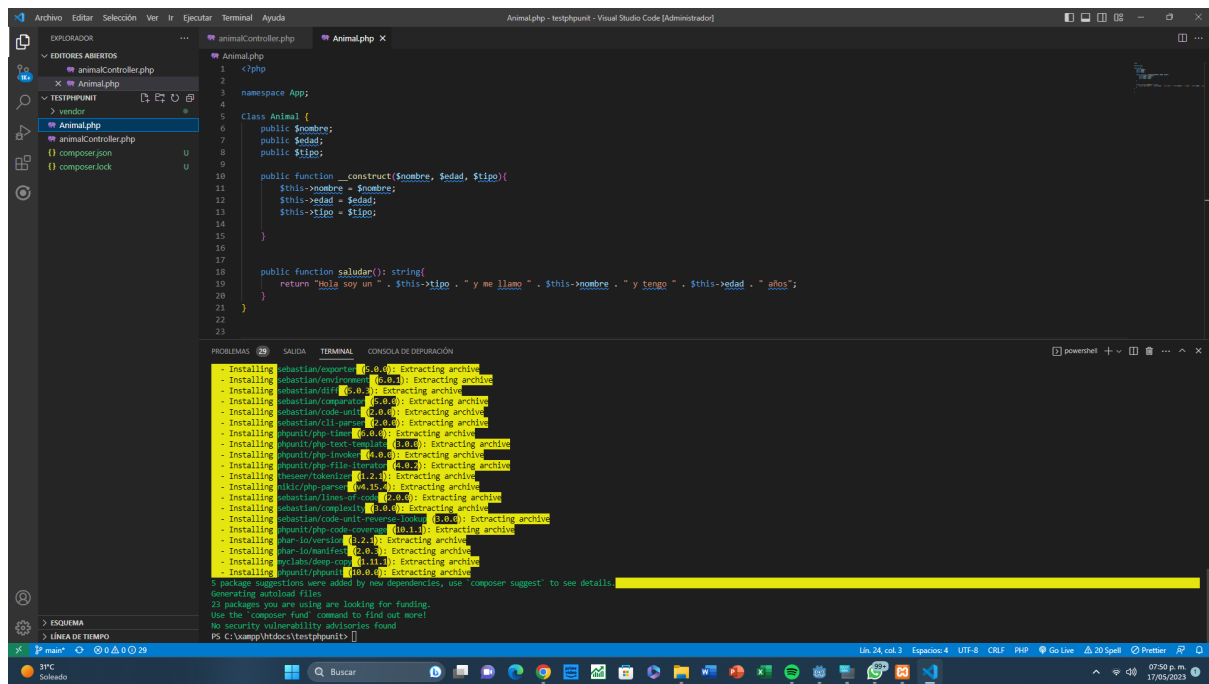
PS C:\xampp\htdocs\testphpunit>
```

Después creamos un archivo llamado “**phpunit.xml**”

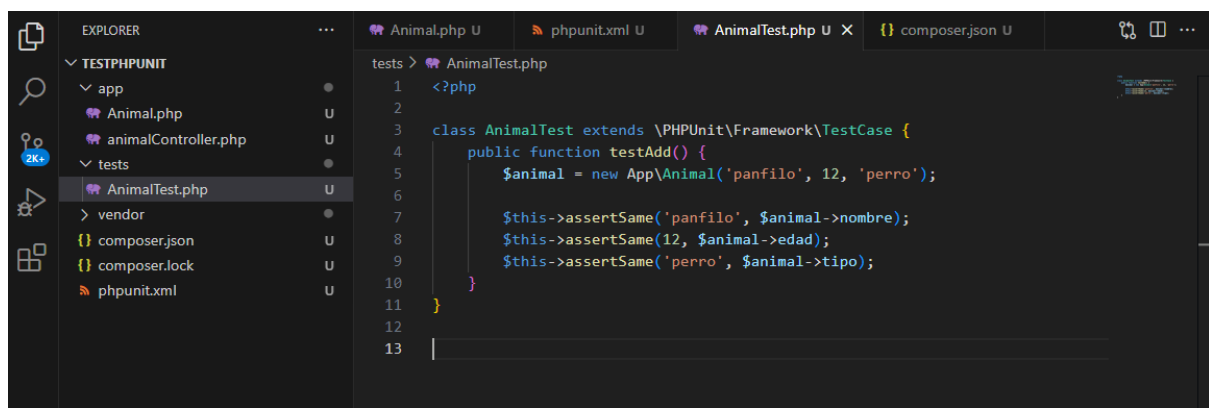


Posteriormente creamos una carpeta llamada tests y en la terminal utilizamos el comando **“vendor/bin/phpunit”**

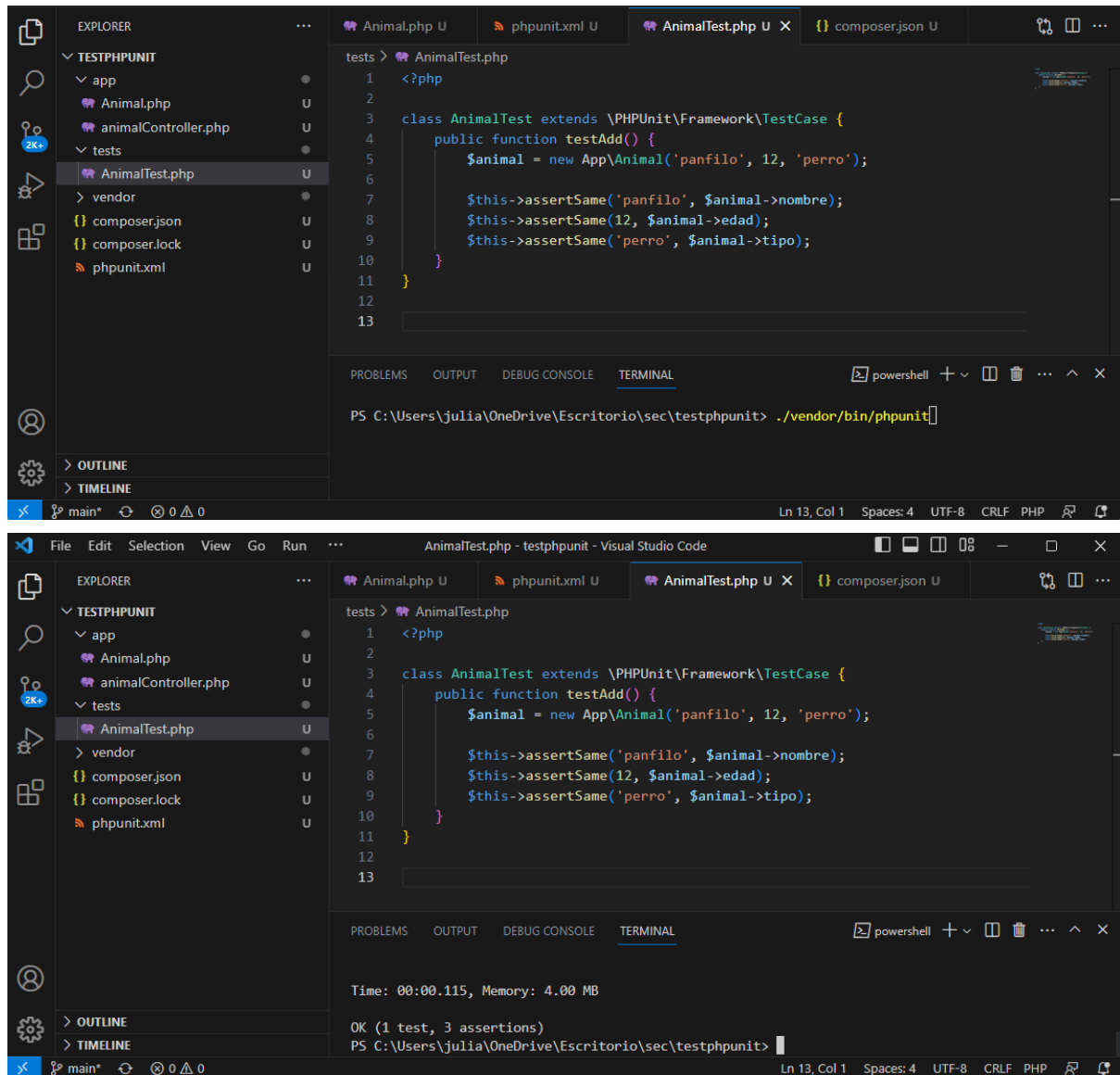




Dentro de la carpeta tests agregamos un documento en este caso **AnimalTest.php** en el cual crearemos el código para realizar las pruebas y configurar el archivo composer para indicar de donde se cargaran las clases.

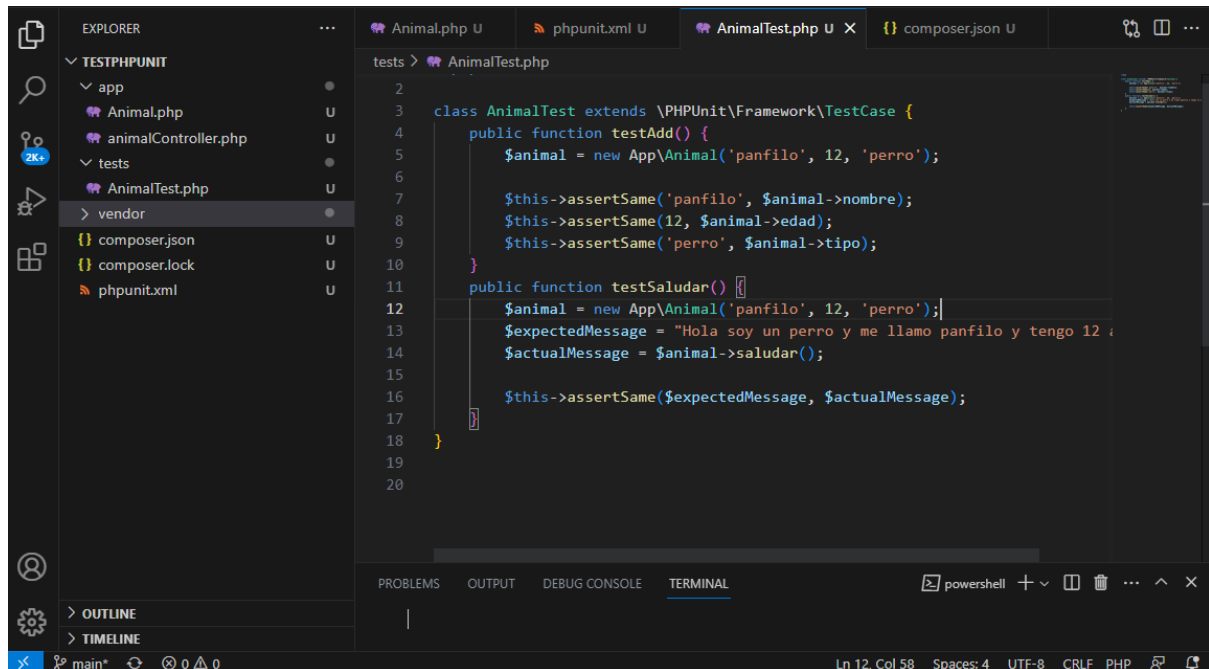


Después de configurar el composer lo que hacemos es en la consola poner el comando “./vendor/bin/phpunit” para que nos muestre el resultado de la prueba.



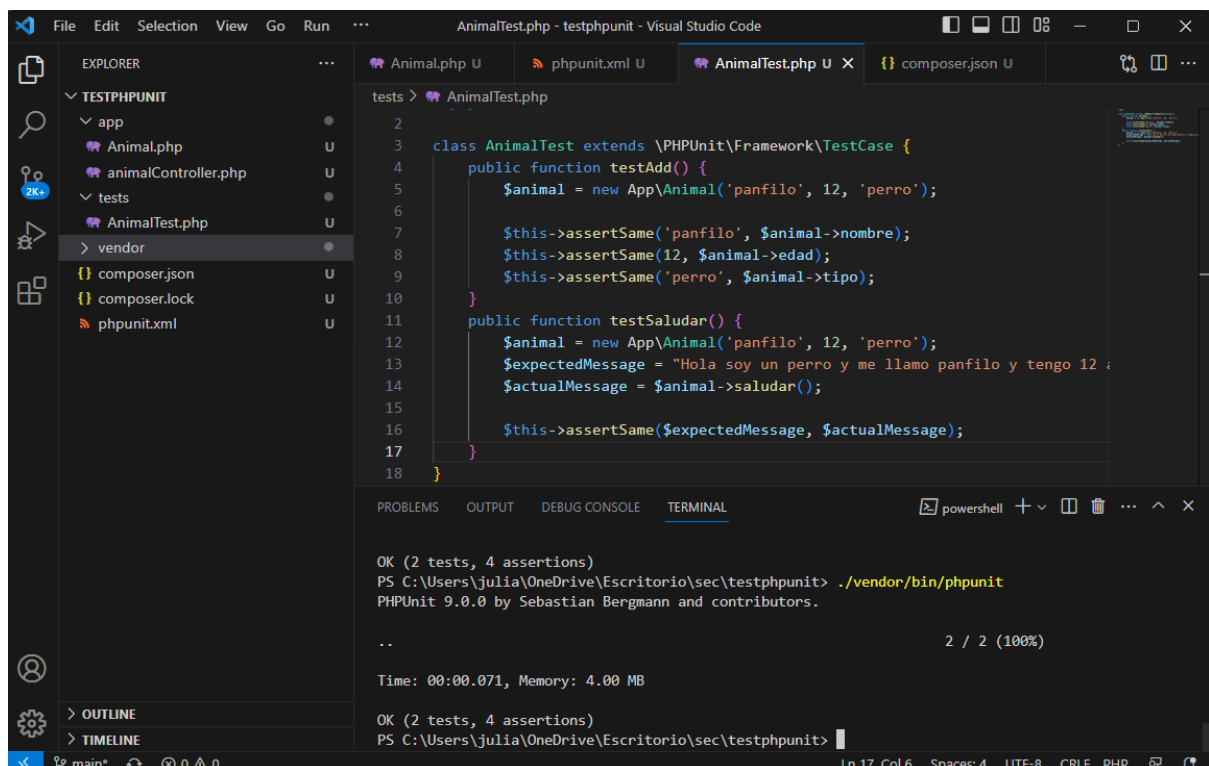


Para testear la función Saludar de la clase Animal se agrega otra función en el archivo **AnimalTest.php** la cual comprueba si el mensaje con los valores que se le dan es igual al mensaje que da la función para eso usamos el comando “./vendor/bin/phpunit” el cual nos dice que no hay errores.



The screenshot shows the Visual Studio Code interface with the Explorer view on the left displaying the project structure. The main editor shows the `AnimalTest.php` file with the following code:

```
2  
3 class AnimalTest extends \PHPUnit\Framework\TestCase {  
4     public function testAdd() {  
5         $animal = new App\Animal('panfilo', 12, 'perro');  
6  
7         $this->assertSame('panfilo', $animal->nombre);  
8         $this->assertSame(12, $animal->edad);  
9         $this->assertSame('perro', $animal->tipo);  
10    }  
11    public function testSaludar() {  
12        $animal = new App\Animal('panfilo', 12, 'perro');  
13        $expectedMessage = "Hola soy un perro y me llamo panfilo y tengo 12 años";  
14        $actualMessage = $animal->saludar();  
15  
16        $this->assertSame($expectedMessage, $actualMessage);  
17    }  
18 }  
19  
20
```



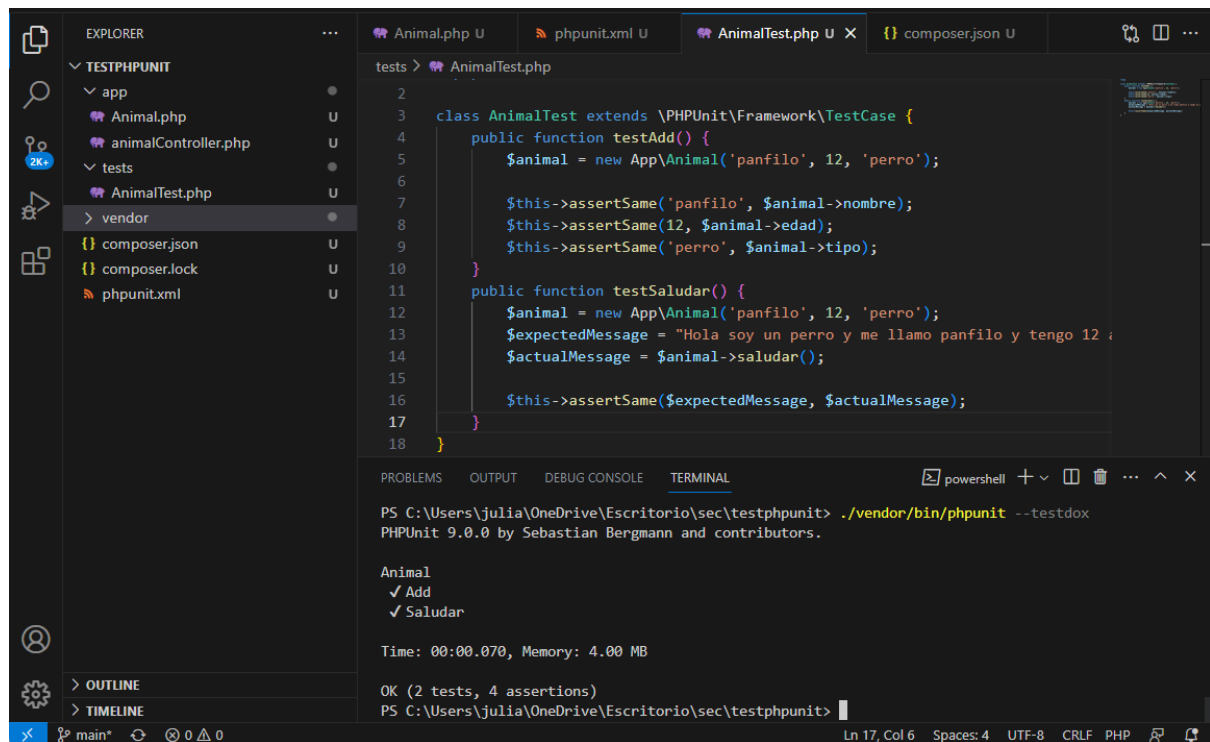
The screenshot shows the Visual Studio Code interface with the Explorer view on the left displaying the project structure. The main editor shows the `AnimalTest.php` file with the following code:

```
3 class AnimalTest extends \PHPUnit\Framework\TestCase {  
4     public function testAdd() {  
5         $animal = new App\Animal('panfilo', 12, 'perro');  
6  
7         $this->assertSame('panfilo', $animal->nombre);  
8         $this->assertSame(12, $animal->edad);  
9         $this->assertSame('perro', $animal->tipo);  
10    }  
11    public function testSaludar() {  
12        $animal = new App\Animal('panfilo', 12, 'perro');  
13        $expectedMessage = "Hola soy un perro y me llamo panfilo y tengo 12 años";  
14        $actualMessage = $animal->saludar();  
15  
16        $this->assertSame($expectedMessage, $actualMessage);  
17    }  
18 }
```

The terminal output shows the results of running the PHPUnit command:

```
OK (2 tests, 4 assertions)  
PS C:\Users\julia\OneDrive\Escritorio\sec\testphpunit> ./vendor/bin/phpunit  
PHPUnit 9.0.0 by Sebastian Bergmann and contributors.  
  
..  
2 / 2 (100%)  
  
Time: 00:00.071, Memory: 4.00 MB  
  
OK (2 tests, 4 assertions)  
PS C:\Users\julia\OneDrive\Escritorio\sec\testphpunit>
```

Resultados de ambas pruebas usando el comando **“vendor/bin/phpunit --testbox”** el cual nos muestra que ambas pruebas se pasaron con éxito.



The screenshot shows the Visual Studio Code interface with the Explorer, Search, and Run and Debug views on the left. The Explorer view shows a project structure with a 'tests' directory containing 'AnimalTest.php'. The Search view shows the contents of 'AnimalTest.php'. The Run and Debug view shows the output of the command `./vendor/bin/phpunit --testbox` in a PowerShell terminal. The output shows that the tests passed successfully.

```
class AnimalTest extends \PHPUnit\Framework\TestCase {
    public function testAdd() {
        $animal = new App\Animal('panfilo', 12, 'perro');

        $this->assertSame('panfilo', $animal->nombre);
        $this->assertSame(12, $animal->edad);
        $this->assertSame('perro', $animal->tipo);
    }

    public function testSaludar() {
        $animal = new App\Animal('panfilo', 12, 'perro');
        $expectedMessage = "Hola soy un perro y me llamo panfilo y tengo 12 años";
        $actualMessage = $animal->saludar();

        $this->assertSame($expectedMessage, $actualMessage);
    }
}
```

```
PS C:\Users\julia\OneDrive\Escritorio\sec\testphpunit> ./vendor/bin/phpunit --testbox
PHPUnit 9.0.0 by Sebastian Bergmann and contributors.

Animal
✓ Add
✓ Saludar

Time: 00:00.070, Memory: 4.00 MB

OK (2 tests, 4 assertions)
PS C:\Users\julia\OneDrive\Escritorio\sec\testphpunit>
```