

Pruebas de Software - PHPUnit

Daniel Felipe Bermeo Torres

Deisy Caterine Gonzalez Bueno

Jennifer Vanessa Mateus Farfán

Servicio Nacional de Aprendizaje

Centro de Electricidad, Electrónica y Telecomunicaciones

2900602 – ADSO

Javier Yara

01-06-2025

Módulo de Productos

```
<?php
use PHPUnit\Framework\TestCase;

require_once __DIR__ . '/../app/Models/Producto.php';
include_once __DIR__ . '/../app/Config/Database.php';

use App\Models\Producto;
use App\Config\Database;

class ProductoTest extends TestCase
{
    protected $producto;

    protected function setUp(): void
    {
        $this->producto = new Producto();
    }

    public function testAñadirProducto()
    {
        // Datos simulados
        $nombreproducto = 'Producto de prueba';
        $precio = 19999;
        $precioproveedor = 199;
        $categoria = 1;
        $estado = 3;
        $foto = __DIR__ . '/imagen_dummy.jpg';

        $tallas = [1];
        $cantidades = [10];
        $colores = ['Rojo'];

        $fechaEntrada = date('Y-m-d');
        $proveedorId = 1;

        // Crear archivo de prueba si no existe
        if (!file_exists($foto)) {
            file_put_contents($foto,
                file_get_contents('https://via.placeholder.com/150'));
        }

        // Ejecutar el método y validar que no falle
        try {
            $this->producto->añadirProducto(

```

```
        $foto,
        $nombreproducto,
        $precio,
        $precioproveedor,
        $colores[0],
        $categoria,
        $estado,
        $tallas,
        $cantidades,
        $colores,
        $fechaEntrada,
        $proveedorId
    );

    $this->assertTrue(true);
} catch (Exception $e) {
    $this->fail("Error al añadir producto: " . $e->getMessage());
}
}

public function testConsultarProducto()
{
    // Obtener todos los productos
    $productos = $this->producto->obtenerProductos();

    $this->assertIsArray($productos);
    $this->assertGreaterThanOrEqual(0, count($productos));

    if (count($productos) > 0) {
        $primerProducto = $productos[0];
        $this->assertArrayHasKey('idProducto', $primerProducto);
        $this->assertArrayHasKey('nombreproducto', $primerProducto);
        $this->assertArrayHasKey('precio', $primerProducto);
        $this->assertArrayHasKey('Detalle_Producto', $primerProducto);
    }
}

public function testActualizarProducto()
{
    $idProducto = 356; // ID de prueba
    $nombreProducto = 'Producto actualizado';
    $precio = 29999;
    $categoria = 1;
    $estado = 3;
```

```

        $resultado = $this->producto->actualizarProducto(
            $idProducto,
            $nombreProducto,
            $precio,
            $categoria,
            $estado
        );

        $this->assertTrue($resultado, 'El producto debería actualizarse
correctamente');
    }

    public function testConsultarTodosLosProductos()
    {
        $productos = $this->producto->obtenerProductos();

        $this->assertIsArray($productos);
        $this->assertGreaterThanOrEqual(0, count($productos));

        if (count($productos) > 0) {
            $primerProducto = $productos[0];
            $this->assertArrayHasKey('idProducto', $primerProducto);
            $this->assertArrayHasKey('nombreProducto', $primerProducto);
            $this->assertArrayHasKey('precio', $primerProducto);
            $this->assertArrayHasKey('Detalle_Producto', $primerProducto);
        }
    }

    public static function tearDownAfterClass(): void
    {
        echo "\n\n*****\n";
        echo "Todas las pruebas del módulo PRODUCTOS se realizaron
correctamente [✓]\n";
        echo "Se validaron agregar, consultar, actualizar y listar
productos.\n";
        echo "*****\n\n";
    }
}

```

Módulo de Proveedores

```
<?php

use PHPUnit\Framework\TestCase;

require_once __DIR__ . '/../app/Models/Proveedor.php';
include_once __DIR__ . '/../app/Config/Database.php';

use App\Models\Proveedor;
use App\Config\Database;

class ProveedorTest extends TestCase
{
    protected $proveedor;

    protected function setUp(): void
    {
        $this->proveedor = new Proveedor();
    }

    public function testAñadirProveedor()
    {
        // Datos simulados para el proveedor
        $nombreproveedor = 'Proveedor Test';
        $telefono = '1234567890';
        $producto = 1; // ID del producto que provee

        try {
            $resultado = $this->proveedor->añadirProveedor(
                $nombreproveedor,
                $telefono,
                $producto
            );

            $this->assertIsInt($resultado);
            $this->assertGreaterThan(0, $resultado);
        } catch (Exception $e) {
            $this->fail("Error al añadir proveedor: " . $e->getMessage());
        }
    }

    public function testConsultarProveedor()
    {
        $resultado = $this->proveedor->consultarProveedor();

        $this->assertIsArray($resultado);
        $this->assertGreaterThanOrEqual(0, count($resultado));
    }
}
```

```
if (count($resultado) > 0) {
    $primerProveedor = $resultado[0];
    $this->assertArrayHasKey('idProveedor', $primerProveedor);
    $this->assertArrayHasKey('nombreproveedor', $primerProveedor);
    $this->assertArrayHasKey('Telefono', $primerProveedor);
    $this->assertArrayHasKey('producto', $primerProveedor);
}
}

public function testBuscarProducto()
{
    $busqueda = 'Proveedor Test';
    $resultado = $this->proveedor->buscarProducto($busqueda);

    $this->assertIsArray($resultado);
    $this->assertGreaterThanOrEqual(0, count($resultado));

    if (count($resultado) > 0) {
        $primerProveedor = $resultado[0];
        $this->assertArrayHasKey('idProveedor', $primerProveedor);
        $this->assertArrayHasKey('nombreproveedor', $primerProveedor);
        $this->assertArrayHasKey('producto', $primerProveedor);
    }
}

public function testActualizarProveedor()
{
    $idProveedor = 1; // Asume que existe un proveedor con ID 1
    $nombreproveedor = 'Proveedor Actualizado';
    $telefono = '9876543210';
    $producto = 1; // Nuevo producto que provee

    $resultado = $this->proveedor->actualizarProveedor(
        $idProveedor,
        $nombreproveedor,
        $telefono,
        $producto
    );

    $this->assertTrue($resultado);
}

public function testConsultarTalla()
{
```

```

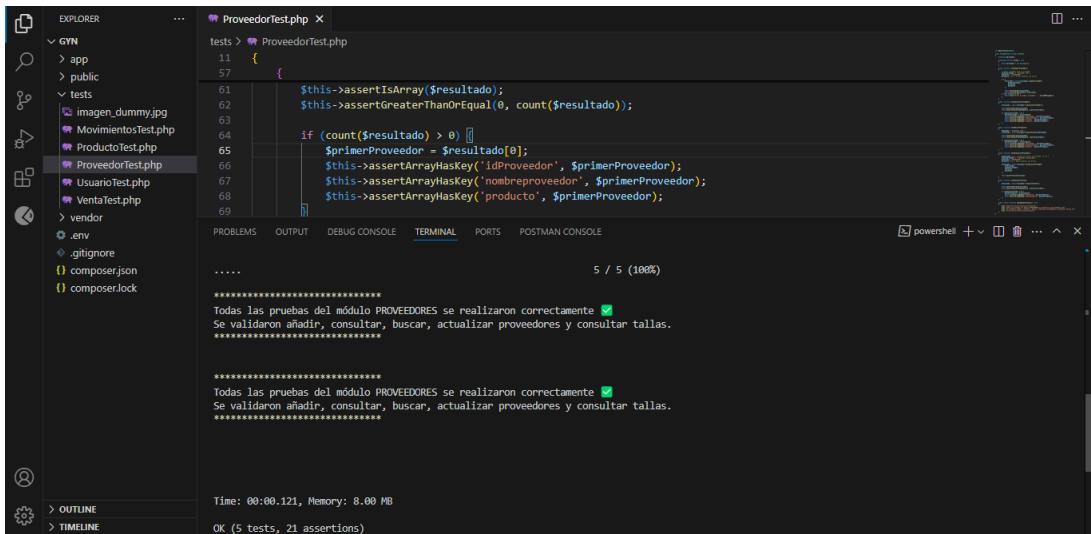
$resultado = $this->proveedor->consultartalla();

$this->assertIsArray($resultado);
$this->assertGreaterThanOrEqual(0, count($resultado));

if (count($resultado) > 0) {
    $primerProducto = $resultado[0];
    $this->assertArrayHasKey('idProducto', $primerProducto);
    $this->assertArrayHasKey('nombreproducto', $primerProducto);
}

public static function tearDownAfterClass(): void
{
    echo "\n\n*****\n";
    echo "Todas las pruebas del módulo PROVEEDORES se realizaron
correctamente ✅\n";
    echo "Se validaron añadir, consultar, buscar, actualizar proveedores
y consultar tallas.\n";
    echo "*****\n\n";
}
}

```



Módulo de Movimientos

```

<?php
use PHPUnit\Framework\TestCase;

require_once __DIR__ . '/../app/Models/Movimientos.php';

```

```
include_once __DIR__ . '/../app/Config/Database.php';

use App\Models\Movimientos;
use App\Config\Database;

class MovimientosTest extends TestCase
{
    protected $movimientos;

    protected function setUp(): void
    {
        $this->movimientos = new Movimientos();
    }

    public function testAñadirMovimiento()
    {
        // Datos simulados para el movimiento
        $entradaProducto = 10;
        $fechaentrada = date('Y-m-d');
        $ProductoidProducto = 1;
        $TallaIdTalla = 1;
        $ProveedoridProveedor = 1;
        $precioproveedor = 10000;
        $anadido = 6;
        $total = 100000;
        $tallas = [1, 2, 3];
        $cantidades = [5, 3, 2];
        $colores = ['Rojo', 'Azul', 'Negro'];

        try {
            $resultado = $this->movimientos->añadirmovimiento(
                $entradaProducto,
                $fechaentrada,
                $ProductoidProducto,
                $TallaIdTalla,
                $ProveedoridProveedor,
                $precioproveedor,
                $anadido,
                $total,
                $tallas,
                $cantidades,
                $colores
            );
        }

        $this->assertIsInt($resultado);
    }
}
```

```

        $this->assertGreaterThan(0, $resultado);
    } catch (Exception $e) {
        $this->fail("Error al añadir movimiento: " . $e->getMessage());
    }
}

public function testObtenerMovimiento()
{
    $resultado = $this->movimientos->obtenerMovimiento();

    $this->assertIsArray($resultado);
    $this->assertGreaterThanOrEqual(0, count($resultado));

    if (count($resultado) > 0) {
        $primerMovimiento = $resultado[0];
        $this->assertArrayHasKey('idProceso', $primerMovimiento);
        $this->assertArrayHasKey('entradaproducto', $primerMovimiento);
        $this->assertArrayHasKey('fecha_entrada', $primerMovimiento);
        $this->assertArrayHasKey('producto', $primerMovimiento);
    }
}

public function testBuscarMovimientos()
{
    $busqueda = 'Proveedor Test';
    $resultado = $this->movimientos->buscarMovimientos($busqueda);

    $this->assertIsArray($resultado);
    $this->assertGreaterThanOrEqual(0, count($resultado));

    if (count($resultado) > 0) {
        $primerMovimiento = $resultado[0];
        $this->assertArrayHasKey('idProceso', $primerMovimiento);
        $this->assertArrayHasKey('proveedor', $primerMovimiento);
    }
}

public static function tearDownAfterClass(): void
{
    echo "\n\n*****\n";
    echo "Todas las pruebas del módulo MOVIMIENTOS se realizaron\n";
    echo "correctamente [✓]\n";
    echo "Se validaron añadir, obtener y buscar movimientos.\n";
    echo "*****\n\n";
}

```

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer:** Shows the project structure under 'GYN' with folders like 'app', 'public', 'tests', and files like 'Imagen_dummy.jpg', 'MovimientosTest.php', 'ProductoTest.php', 'ProveedorTest.php', 'UsuarioTest.php', and 'VentaTest.php'. It also lists 'vendor' and configuration files like '.env', '.gitignore', 'composer.json', and 'composer.lock'.
- Editor:** Displays the content of 'MovimientosTest.php'. The code includes imports for PHPUnit and App\Models\Movimientos, and defines a class MovimientosTest extending TestCase. The editor shows the first 10 lines of the file.
- Terminal:** Shows the command PS C:\xampp\htdocs\GYN\ ./vendor/bin/phpunit tests/MovimientosTest.php being run. The output indicates PHPUnit 11.5.22 was used by Sebastian Bergmann and contributors, with a runtime of PHP 8.2.12. The test results show 3 / 3 (100%) tests passed, with validation messages for adding, getting, and searching movements.

Módulo de Ventas

```
<?php
use PHPUnit\Framework\TestCase;

require_once __DIR__ . '/../app/Models/Venta.php';
include_once __DIR__ . '/../app/Config/Database.php';

use App\Models\Venta;
use App\Config\Database;

class VentaTest extends TestCase
{
    protected $venta;

    protected function setUp(): void
    {
        $this->venta = new Venta();
    }

    public function testAgregarVenta()
    {
        // Datos simulados para la venta
        $fechaventa = date('Y-m-d H:i:s');
        $id_estado = 1; // Estado activo
        $productos = [
            [
                'id_producto' => 1,
                'cantidad' => 1,
                'precio' => 100.00
            ],
            [
                'id_producto' => 2,
                'cantidad' => 1,
                'precio' => 200.00
            ]
        ];
        $venta->fechaventa = $fechaventa;
        $venta->id_estado = $id_estado;
        $venta->productos = $productos;
        $venta->guardar();
    }
}
```

```

        'idProducto' => 1,
        'talla' => 1,
        'cantidad' => 2,
        'valorunitario' => 50000,
        'cliente' => 1
    ]
];
$documento = '1140914079';

try {
    $idFactura = $this->venta->agregarVenta(
        $fechaventa,
        $id_estado,
        $productos,
        $documento
    );

    $this->assertIsInt($idFactura);
    $this->assertGreaterThan(0, $idFactura);
} catch (Exception $e) {
    $this->fail("Error al agregar venta: " . $e->getMessage());
}
}

public function testObtenerVentas()
{
    $resultado = $this->venta->obtenerVentas();

    $this->assertIsArray($resultado);
    $this->assertGreaterThanOrEqual(0, count($resultado));

    if (count($resultado) > 0) {
        $primeraVenta = $resultado[0];
        $this->assertArrayHasKey('idFactura', $primeraVenta);
        $this->assertArrayHasKey('fechaventa', $primeraVenta);
        $this->assertArrayHasKey('total', $primeraVenta);
        $this->assertArrayHasKey('productos', $primeraVenta);
    }
}

public function testBuscarVentas()
{
    $busqueda = "1"; // Buscar venta con ID 1
    $resultado = $this->venta->buscarVentas($busqueda);
}

```

```

    $this->assertIsArray($resultado);
    $this->assertGreaterThanOrEqual(0, count($resultado));

    if (count($resultado) > 0) {
        $venta = $resultado[0];
        $this->assertArrayHasKey('idFactura', $venta);
        $this->assertArrayHasKey('fechaventa', $venta);
        $this->assertArrayHasKey('total', $venta);
        $this->assertArrayHasKey('productos', $venta);
    }
}

public function testPagarVenta()
{
    $idFactura = 1; // Asume que existe una venta con ID 1
    $id_estadof = 1;
    $fechaventa = date('Y-m-d H:i:s');

    $resultado = $this->venta->pagarVenta($idFactura, $id_estadof,
$fechaventa);
    $this->assertTrue($resultado);
}

public function testNoPagarVenta()
{
    $idFactura = 1; // Asume que existe una venta con ID 1
    $id_estadof = 2;
    $fechaventa = date('Y-m-d H:i:s');

    $resultado = $this->venta->nopagarVenta($idFactura, $id_estadof,
$fechaventa);
    $this->assertTrue($resultado);
}

public static function tearDownAfterClass(): void
{
    echo "\n\n*****\n";
    echo "Todas las pruebas del módulo VENTAS se realizaron
correctamente \n";
    echo "Se validaron agregar, obtener, buscar, pagar y no pagar
ventas.\n";
    echo "*****\n\n";
}
}

```

File Edit Selection View Go Run ... GYN

EXPLORER
GYN
tests > VentaTest.php

```
11 {
12     ...
13 }
14     public function testAgregarVenta()
15     {
16         // Datos simulados para la venta
17         $fechaventa = date('Y-m-d H:i:s');
18         $id_estadof = 1; //Estado activo
19     }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

PS C:\xampp\htdocs\GYN> ./Vendor/bin/phpunit tests/VentaTest.php
PHPUnit 11.5.22 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.12

.....

Todas las pruebas del módulo VENTAS se realizaron correctamente ✓

Se validaron agregar, obtener, buscar, pagar y no pagar ventas.

Time: 00:00.039, Memory: 8.00 MB

OK (5 tests, 18 assertions)

PS C:\xampp\htdocs\GYN> []