



## ***Pruebas de Software PHPUnit***

### **PATY SPORT**

#### **Preparado Por:**

Lilley Caicedo Nieto  
Angel Daniel Puentes Ardila  
Jonny Alejandro Pulido Escobar  
Nayarit Glet Gutierrez Nieto

#### **Fecha de Creación:**

19 de Mayo del 2025

**Versión [1.0]**

**Estado [Borrador/En Revisión/Aprobado]**

**CEET**

(CENTRO DE ELECTRICIDAD ELECTRONICA Y  
TELECOMUNICACIONES)

**TRIMESTRE VI**



## Documentación de Pruebas del Módulo de Productos con PHPUnit

### 1. Introducción

Este documento describe el proceso de integración de **PHPUnit** en el proyecto PATY SPORT desarrollado en PHP puro, así como las pruebas realizadas para el módulo de gestión de productos, usuarios, ticket de salida y ticket de ingreso. Se detalla cómo se configuró PHPUnit, cómo se ejecutaron las pruebas y una explicación de cada prueba realizada para los métodos del módulo productos.

### 2. ¿Qué es PHPUnit?

PHPUnit es un framework para realizar pruebas unitarias en PHP. Permite automatizar la ejecución de pruebas, asegurando que las funciones y métodos del código funcionen correctamente y se mantengan estables frente a cambios futuros.

### 3. Integración de PHPUnit en un Proyecto PHP Puro

#### 3.1. Instalación de PHPUnit

Para integrar PHPUnit en un proyecto PHP sin frameworks, se siguieron los siguientes pasos:

##### 1. Agregar PHPUnit como dependencia de desarrollo

En la raíz del proyecto, ejecutar:

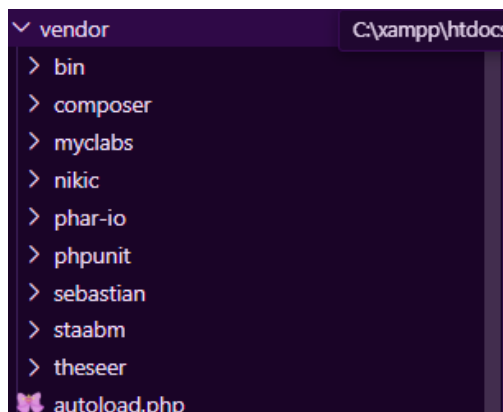
`composer require --dev phpunit/phpunit`

```
PS C:\xampp\htdocs\Bootstrap> composer require --dev phpunit/phpunit
Warning: PHPUnit 11.5.20 requires PHP 8.1 or higher, which is not installed on your system. You can ignore this message and run PHPUnit, but some features may not work correctly.
Running composer update phpunit/phpunit
Loading composer repositories with package information
Updating dependencies
Lock file operations: 27 installs, 0 updates, 0 removals
  - Locking myclabs/deep-copy (1.13.1)
  - Locking nikic/php-parser (v5.4.0)
  - Locking phar-io/manifest (2.0.4)
  - Locking phar-io/version (3.2.1)
  - Locking phpunit/php-code-coverage (11.0.9)
  - Locking phpunit/php-file-iterator (5.1.0)
  - Locking phpunit/php-invoker (5.0.1)
  - Locking phpunit/php-text-template (4.0.1)
  - Locking phpunit/php-timer (7.0.1)
  - Locking phpunit/phpunit (11.5.20)
  - Locking sebastian/cli-parser (1.0.2)
  - Locking sebastian/code-unit (3.0.3)
  - Locking sebastian/code-unit-reverse-lookup (4.0.1)
  - Locking sebastian/comparator (6.3.1)
  - Locking sebastian/complexity (4.0.1)
  - Locking sebastian/diff (6.0.2)
  - Locking sebastian/environment (7.2.0)
  - Locking sebastian/exporter (6.3.0)
  - Locking sebastian/global-state (7.0.2)
  - Locking sebastian/lines-of-code (3.0.1)
  - Locking sebastian/object-enumerator (6.0.1)
  - Locking sebastian/object-reflector (4.0.1)
  - Locking sebastian/recursion-context (6.0.2)
  - Locking sebastian/type (5.1.2)
  - Locking sebastian/version (5.0.2)
  - Locking staabm/side-effects-detector (1.0.5)
  - Locking theseer/tokenizer (1.2.3)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 27 installs, 0 updates, 0 removals
  - Downloading phpunit/phpunit (11.5.20)
  - Installing staabm/side-effects-detector (1.0.5): Extracting archive
  - Installing sebastian/version (5.0.2): Extracting archive
  - Installing sebastian/type (5.1.2): Extracting archive
  - Installing sebastian/text-template (4.0.1): Extracting archive
  - Installing sebastian/recursion-context (6.0.2): Extracting archive
  - Installing sebastian/object-reflector (4.0.1): Extracting archive
  - Installing sebastian/global-state (7.0.2): Extracting archive
  - Installing sebastian/exporter (6.3.0): Extracting archive
  - Installing sebastian/environment (7.2.0): Extracting archive
  - Installing sebastian/diff (6.0.2): Extracting archive
  - Installing sebastian/comparator (6.3.1): Extracting archive
  - Installing sebastian/code-unit (3.0.3): Extracting archive
```



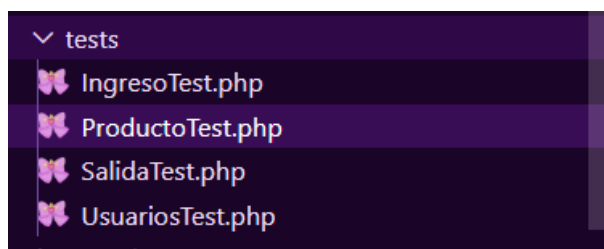
## 2. Se crea automáticamente la carpeta **VENDOR**

Se crea en el momento de que termine la ejecución del comando dado, la carpeta vendor que es donde se realizan las pruebas.

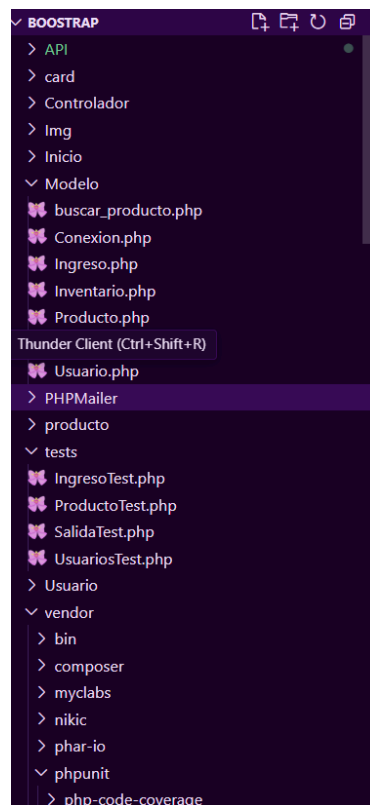


## 3. Crear la carpeta de pruebas

Se creó una carpeta llamada tests/ en la raíz del proyecto principal para almacenar los archivos de prueba.



## 3.2. Estructura del Proyecto con PHPUnit





## 4. Explicación de las Pruebas del Módulo Productos (Método por Método)

### 4.1. testAgregarProducto()

- **Qué prueba:**  
Verifica que el método agregarProducto() pueda insertar un producto sin lanzar excepciones.
- **Cómo:**  
Se llama a AgregarProducto() con datos de prueba válidos.  
No se espera un valor de retorno, por eso la prueba es que no lance excepción.

```
public function testAgregarProducto()
{
    $nombre = 'Producto de prueba Agregar Producto';
    $precio = 199.99;
    $iva = 19;
    $imagen = 'imagen.jpg';
    $id_categoria = 121;
    $id_estado = 101;
    $descripcion = 'Descripción del producto';
    $id_marca = 30;
    $id_talla = 201;
    $documento_usuario = '505547805';

    try {
        $this->producto->agregarProducto(
            $nombre,
            $precio,
            $iva,
            $imagen,
            $id_categoria,
            $id_estado,
            $descripcion,
            $id_marca,
            $id_talla,
            $documento_usuario
        );
        $this->assertTrue(true);
    } catch (\Exception $e) {
        $this->fail('El método agregarProducto lanzó una excepción: ' . $e->getMessage());
    }
}
```

	CodigoProducto	Nombre	Precio	Stock	IVA	Imagen	CategoriaCodigoCategoria	EstadoCodigoEstado	Descripcion	MarcasCodigoMarca	Tallas
<input type="checkbox"/>	340	Sudadera de Caballero Lilley	25000	50	19	NULL	124	102	Sudadera encauchada en pretina y bota		38
<input type="checkbox"/>	356	Producto actualizado Lilley	299	NULL	19	imagen.jpg	121	101	Producto actualizado		30
<input type="checkbox"/>	357	Producto de prueba Agregar Producto	199	NULL	19	imagen.jpg	121	101	Descripción del producto		30

- **Validación:**  
Si no se lanza ninguna excepción, la prueba pasa.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

OK (5 tests, 12 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/ProductoTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

....                                                4 / 4 (100%)

*****
Todas las pruebas del módulo PRODUCTOS se realizaron correctamente ✓
Se validaron agregar, consultar, actualizar y listar productos.
*****

Time: 00:00.083, Memory: 8.00 MB

OK (4 tests, 11 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap> █
```

## 4.2. testConsultarProducto()

- **Qué prueba:**

Comprueba que el método consultarProducto() retorne los datos correctos de un producto específico.

- **Cómo:**

Primero inserta un producto con agregarProducto().

Luego obtiene todos los productos y selecciona el último insertado.

Finalmente, consulta el producto por su ID para validar que los datos coinciden.

```
public function testConsultarProducto()
{
    // Insertar un producto antes de consultarlo
    $this->producto->agregarProducto(
        'Producto test consulta de producto',
        100.0,
        19,
        'test.jpg',
        121,
        101,
        'Producto de prueba para consulta',
        30,
        201,
        '505547805'
    );

    // Obtener el último producto insertado (suponiendo que consultarProducto sin parámetro devuelve todos)
    $todos = $this->producto->consultarProducto(); // Asumiendo devuelve mysqli_result
    $ultimoProducto = null;
    while ($fila = $todos->fetch_assoc()) {
        $ultimoProducto = $fila;
    }

    $this->assertNotNull($ultimoProducto, 'Debe existir al menos un producto para probar');

    // Consultar el producto por su ID real
    $resultado = $this->producto->consultarProducto($ultimoProducto['CodigoProducto']);

    $this->assertInstanceOf(mysqli_result::class, $resultado);
    $this->assertTrue($resultado->num_rows > 0, 'Debe existir al menos un producto con ese ID');

    $producto = $resultado->fetch_assoc();
    $this->assertArrayHasKey('CodigoProducto', $producto, 'Debe contener el campo CodigoProducto');
    $this->assertArrayHasKey('Nombre', $producto, 'Debe contener el campo Nombre');
}
```



Servidor 127.0.0.1 » Base de datos: paty\_sport » Tabla: producto

Examinar

Estructura

SQL

Buscar

Insertar

Exportar

Importar

Privilegios

Operaciones

Disparadores

Opciones extra

<

- **Validación:**

El resultado debe ser un objeto `mysqli_result` con al menos una fila que contenga las claves esperadas (por ejemplo, `CodigoProducto` y `Nombre`).

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

OK (5 tests, 12 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/ProductoTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

.....                                     4 / 4 (100%)

*****
Todas las pruebas del módulo PRODUCTOS se realizaron correctamente ✓
Se validaron agregar, consultar, actualizar y listar productos.
*****

Time: 00:00.083, Memory: 8.00 MB

OK (4 tests, 11 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```

#### 4.3. testActualizarProducto()

- **Qué prueba:**

Verifica que el método `actualizarProducto()` modifique correctamente los datos de un producto existente.

- **Cómo:**

Se llama a `actualizarProducto()` con un ID de producto existente y nuevos datos.



```
public function testActualizarProducto()  
{  
    $id_producto = 356; // Asume que el producto con ID 1 existe  
    $nombre = 'Producto actualizado Lilley';  
    $precio = 299.99;  
    $iva = 19;  
    $id_categoria = 121;  
    $id_estado = 101;  
    $descripcion = 'Producto actualizado';  
    $id_marca = 30;  
    $id_talla = 201;  
    $documento_usuario = '1080180837';  
  
    $resultado = $this->producto->actualizarProducto(  
        $id_producto,  
        $nombre,  
        $precio,  
        $iva,  
        $id_categoria,  
        $id_estado,  
        $descripcion,  
        $id_marca,  
        $id_talla,  
        $documento_usuario  
    );  
  
    $this->assertTrue($resultado, 'El producto debería actualizarse correctamente');  
}
```

Servidor: 127.0.0.1 » Base de datos: paty\_sport » Tabla: producto

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Disparadores

Opciones extra

		CodigoProducto	Nombre	Precio	Stock	IVA	Imagen	CategoriaCodigoCategoria	EstadoCodigoEstado	Descripcion	MarcasCodigoMarca	Tallas
<input type="checkbox"/>	Editar Copiar Borrar	340	Sudadera de Caballero Lilley	25000	50	19	NULL	124	102	Sudadera encauchada en pretina y bota		38
<input type="checkbox"/>	Editar Copiar Borrar	356	Producto actualizado Lilley	299	NULL	19	imagen.jpg	121	101	Producto actualizado		30
<input type="checkbox"/>	Editar Copiar Borrar	357	Producto de prueba Agregar Producto	199	NULL	19	imagen.jpg	121	101	Descripción del producto		30
<input type="checkbox"/>	Editar Copiar Borrar	358	Producto test consulta de producto	100	NULL	19	test.jpg	121	101	Producto de prueba para consulta		30

Selecciónar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

<< < 2 > >> Mostrando todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Operaciones sobre los resultados de la consulta

Imprimir Copiar al portapapeles Exportar Mostrar gráfico Crear vista

- Validación:

El método debe retornar true, indicando que la actualización fue exitosa.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

OK (5 tests, 12 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/ProductoTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

....                                                4 / 4 (100%)

*****
Todas las pruebas del módulo PRODUCTOS se realizaron correctamente ✓
Se validaron agregar, consultar, actualizar y listar productos.
*****

Time: 00:00.083, Memory: 8.00 MB

OK (4 tests, 11 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```

## 5. Consideraciones Adicionales

- **Dependencias de Datos:**

Para que las pruebas funcionen correctamente, los IDs usados en las pruebas (id\_categoria, id\_estado, id\_marca, id\_talla, y documento\_usuario) deben existir en sus respectivas tablas de la base de datos.

- **Configuración de Base de Datos:**

Se recomienda tener una base de datos de pruebas para no afectar los datos reales del proyecto.

- **Método tearDownAfterClass():**

Al finalizar todas las pruebas, imprime un mensaje en consola indicando que todas las pruebas del módulo productos se realizaron con éxito.

```
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";
}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

OK (5 tests, 12 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/ProductoTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

....                                                4 / 4 (100%)

*****
Todas las pruebas del módulo PRODUCTOS se realizaron correctamente ✓
Se validaron agregar, consultar, actualizar y listar productos.
*****

Time: 00:00.083, Memory: 8.00 MB

OK (4 tests, 11 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```





## 6. Ejecución y Resultado de las Pruebas

Al ejecutar `vendor/bin/phpunit` en la raíz del proyecto, se corren automáticamente todas las pruebas definidas en `ProductoTest.php`. Si todas las pruebas pasan, el resultado será algo similar a:

```
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit
tests/ProductoTest.php
```

```
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.
```

```
Runtime:      PHP 8.2.12
```

```
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.
```

```
Runtime:      PHP 8.2.12
```

```
.... 4 / 4 (100%)
```

```
*****
```

Todas las pruebas del módulo PRODUCTOS se realizaron correctamente ✓

Se validaron agregar, consultar, actualizar y listar productos.

```
*****
```

```
Time: 00:00.083, Memory: 8.00 MB
```

```
OK (4 tests, 11 assertions)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

OK (5 tests, 12 assertions)
● PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/ProductoTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

.... 4 / 4 (100%)

*****
Todas las pruebas del módulo PRODUCTOS se realizaron correctamente ✓
Se validaron agregar, consultar, actualizar y listar productos.
*****

Time: 00:00.083, Memory: 8.00 MB

OK (4 tests, 11 assertions)
○ PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```



## 7. Ejecución de las Pruebas

Para ejecutar las pruebas, en la terminal se debe correr el comando:

**vendor/bin/phpunit tests/ProductoTest.php**

A screenshot of a terminal window with a dark background and a purple header bar containing tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and POSTMAN CONSOLE. The TERMINAL tab is active. The output shows a successful test run: "OK (5 tests, 12 assertions)" followed by a command prompt "PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/ProductoTest.php".

Este comando ejecuta todas las pruebas dentro del directorio /tests y muestra un reporte con resultados.

A screenshot of a terminal window showing the output of the PHPUnit command. The output includes the same "OK (5 tests, 12 assertions)" message, followed by the command prompt and the command executed. Below this, it shows "PHPUnit 11.5.20 by Sebastian Bergmann and contributors." and "Runtime: PHP 8.2.12". A progress bar shows "4 / 4 (100%)". A message in Spanish states: "Todas las pruebas del módulo PRODUCTOS se realizaron correctamente" followed by a green checkmark icon. Below this, it says "Se validaron agregar, consultar, actualizar y listar productos." and "Time: 00:00.083, Memory: 8.00 MB". At the bottom, it shows "OK (4 tests, 11 assertions)" and the command prompt.



## Documentación de Pruebas del Módulo de Usuarios con PHPUnit

### 1. Introducción

Este documento describe el proceso de integración de **PHPUnit** en el proyecto PATY SPORT desarrollado en PHP puro, así como las pruebas realizadas para el módulo de gestión de usuarios. Se detalla cómo se configuró PHPUnit, cómo se ejecutaron las pruebas y una explicación de cada prueba realizada para los métodos del módulo productos.

### 2. Explicación de las Pruebas del Módulo Productos (Método por Método)

#### 3.1. testAgregarUsuario()

- **Qué prueba:**  
Verifica que el método agregarUsuario() pueda insertar un usuario sin lanzar excepciones.
- **Cómo:**  
Se llama a AgregarUsuario() con datos de prueba válidos.  
No se espera un valor de retorno, por eso la prueba es que no lance excepción.

```
public function testAgregarUsuario()
{
    $tipo_documento = 'CC';
    $num_documento = '1080837';
    $nombres = 'Nombre de prueba lilley';
    $apellidos = 'Apellido de prueba';
    $telefono = '3101234567';
    $correo = 'prueba@example.com';
    $rol_id = 22;
    $estado_codigo = 101;
    $genero_id = 2;

    try {
        $this->usuario->agregarUsuario(
            $tipo_documento,
            $num_documento,
            $nombres,
            $apellidos,
            $telefono,
            $correo,
            $rol_id,
            $estado_codigo,
            $genero_id
        );
        $this->assertTrue(true, 'El usuario debería agregarse correctamente');
    } catch (\Exception $e) {
        $this->fail('El método agregarUsuario lanzó una excepción: ' . $e->getMessage());
    }
}
```



Servidor: 127.0.0.1 » Base de datos: paty_sport » Tabla: usuario										
Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Disparadores										
		Tipo_Documento	Num_Documento	Nombres	Apellidos	Teléfono	Correo	RolidRol	EstadoCodigoEstado	GeneroidGenero
<input type="checkbox"/>	Editar Copiar Borrar	C.C	202327	MIA	Nieto	38456468	johannietooo764@gmail.com	NULL	NULL	NULL
<input type="checkbox"/>	Editar Copiar Borrar	C.E	283745	María	caicedo	2147483647	niet@gmail.com	25	101	1
<input type="checkbox"/>	Editar Copiar Borrar	T.I	654665	Mafe	Utierrez	2147483647	Ndjahhaha@gmail.com	NULL	NULL	NULL
<input type="checkbox"/>	Editar Copiar Borrar	NULL	656565	cinco	seis	66555	sesenta@gmail.com	NULL	101	1
<input type="checkbox"/>	Editar Copiar Borrar	PPT	784514	Dilan Sebastian	Parras Pinoza	2147483647	Almamodafres@gmail.com	24	101	1
<input type="checkbox"/>	Editar Copiar Borrar	T.I	845899	Lily	Nieto	2147483647	Ndjahhaha@gmail.com	NULL	NULL	NULL
<input type="checkbox"/>	Editar Copiar Borrar	C.C	875643	Yonathan	iyutrewq	2147483647	laco@gmail.com	25	101	4
<input type="checkbox"/>	Editar Copiar Borrar	CC	1080837	Nombre de prueba Lilley	Apellido de prueba	2147483647	prueba@example.com	22	101	2
<input type="checkbox"/>	Editar Copiar Borrar	NULL	7042025	Pandita	Gutierrez	45334353	pandita@gmail.com	NULL	101	1
<input type="checkbox"/>	Editar Copiar Borrar	C.C	9346500	Javier	Yara	2147483647	gil@gmail.com	25	101	2
↑ <input type="checkbox"/> Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar										
1 > >> <input type="checkbox"/> Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna										
Operaciones sobre los resultados de la consulta										
Imprimir Copiar al portapapeles Exportar Mostrar gráfico Crear vista										

- **Validación:**  
Si no se lanza ninguna excepción, la prueba pasa.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/UsuariosTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

.....                                     5 / 5 (100%)

*****
Todas las pruebas del módulo USUARIOS se realizaron correctamente ✓
Se validaron agregar, consultar, actualizar, así como consultar roles, estados y géneros.
*****

Time: 00:00.120, Memory: 8.00 MB

OK (5 tests, 13 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```

### 3.2. testActualizarUsuario()

- **Qué prueba:**  
Verifica que el método actualizarUsuario() modifique correctamente los datos de un usuario existente.



- **Cómo:**

Se llama a actualizarUsuario() con un número de documento del usuario existente y nuevos datos.

```
public function testActualizarUsuarioExistente()
{
    $num_documento_actualizar = '112233445';
    $nuevo_correo = 'actualizado.nuevo@example.com';

    $resultado = $this->usuario->actualizarUsuario(
        'CE',
        $num_documento_actualizar,
        $nuevos_nombres,
        'Apellido Actualizar',
        '3155555555',
        $nuevo_correo,
        22,
        2
    );

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

    // Verificar que la actualización se realizó correctamente consultando el usuario
    $usuario_actualizado_resultado = $this->usuario->consultarUsuario($num_documento_actualizar);
    $usuario_actualizado = $usuario_actualizado_resultado->fetch_assoc();
    $this->assertEquals($nuevos_nombres, $usuario_actualizado['Nombres'], 'El nombre del usuario debería haberse actualizado');
    $this->assertEquals($nuevo_correo, $usuario_actualizado['Correo'], 'El correo del usuario debería haberse actualizado');
}
```

	Tipo_Documento	Num_Documento	Nombres	Apellidos	Telefono	Correo	RolIdRol	EstadoCodigoEstado	GeneroidGe
<input type="checkbox"/>	Editar Copiar Borrar C.C	93481145	Javier	Yara	2147483647	javier@gmail.com	NULL	NULL	
<input type="checkbox"/>	Editar Copiar Borrar C.C	93481465	5javier	6yara	2147483647	javier@gmail.com	25	101	
<input type="checkbox"/>	Editar Copiar Borrar C.C	100609905	Yaneth	Nieto	68338245	nietolily4@gmail.com	NULL	NULL	
<input type="checkbox"/>	Editar Copiar Borrar CC	108042789	Nombre de prueba Lilley	Apellido de prueba	2147483647	prueba@example.com	22	101	
<input type="checkbox"/>	Editar Copiar Borrar CC	108082789	Nombre de prueba Lilley	Apellido de prueba	2147483647	prueba@example.com	22	101	
<input type="checkbox"/>	Editar Copiar Borrar CE	112233445	Usuario Actualizado Nuevo	Apellido Actualizar	2147483647	actualizado.nuevo@example.com	22	101	
<input type="checkbox"/>	Editar Copiar Borrar CC	123432789	Nombre de prueba	Apellido de prueba	2147483647	prueba@example.com	22	101	
<input type="checkbox"/>	Editar Copiar Borrar C.C	123456789	Andres	Mullo	1234567	andres	NULL	NULL	
<input type="checkbox"/>	Editar Borrar NULL	190831983	Andres	Caicedo Sanchez	464646	gil@gmail.com	NULL	101	

- **Validación:**

El método debe retornar true, indicando que la actualización fue exitosa.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/UsuariosTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:    PHP 8.2.12

.....                                         5 / 5 (100%)

*****
Todas las pruebas del módulo USUARIOS se realizaron correctamente ✓
Se validaron agregar, consultar, actualizar , así como consultar roles, estados y géneros.
*****

Time: 00:00.120, Memory: 8.00 MB

OK (5 tests, 13 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```

#### 4. Consideraciones Adicionales

- **Dependencias de Datos:**  
Para que las pruebas funcionen correctamente, los Números de documentos usados en las pruebas (id\_rol, id\_estado y id\_genero) deben existir en sus respectivas tablas de la base de datos.
- **Configuración de Base de Datos:**  
Se recomienda tener una base de datos de pruebas para no afectar los datos reales del proyecto.
- **Método tearDownAfterClass():**  
Al finalizar todas las pruebas, imprime un mensaje en consola indicando que todas las pruebas del módulo usuario se realizaron con éxito.

```
public static function tearDownAfterClass(): void
{
    echo "\n\n*****\n";
    echo "Todas las pruebas del módulo USUARIOS se realizaron correctamente ✓\n";
    echo "Se validaron agregar, consultar, actualizar , así como consultar roles, estados y géneros.\n";
    echo "*****\n\n";
}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/UsuariosTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:    PHP 8.2.12

.....                                         5 / 5 (100%)

*****
Todas las pruebas del módulo USUARIOS se realizaron correctamente ✓
Se validaron agregar, consultar, actualizar , así como consultar roles, estados y géneros.
*****

Time: 00:00.120, Memory: 8.00 MB

OK (5 tests, 13 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```



## 5. Ejecución y Resultado de las Pruebas

Al ejecutar `vendor/bin/phpunit` en la raíz del proyecto, se corren automáticamente todas las pruebas definidas en `UsuariosTest.php`. Si todas las pruebas pasan, el resultado será algo similar a:

```
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit
tests/UsuariosTest.php
```

PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.12

..... 5 / 5 (100%)

\*\*\*\*\*

Todas las pruebas del módulo USUARIOS se realizaron correctamente ✓

Se validaron agregar, consultar, actualizar , así como consultar roles, estados y géneros.

\*\*\*\*\*

Time: 00:00.120, Memory: 8.00 MB

OK (5 tests, 13 assertions)

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

PHPUnit 11.5.20 by Sebastian Bergmann and contributors.
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/UsuariosTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.12

O
..... 5 / 5 (100%)
..... 5 / 5 (100%)

*****
Todas las pruebas del módulo USUARIOS se realizaron correctamente ✓
Se validaron agregar, consultar, actualizar , así como consultar roles, estados y géneros.
*****
*****

Time: 00:00.120, Memory: 8.00 MB

OK (5 tests, 13 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap> |
```

## 7. Ejecución de las Pruebas

Para ejecutar las pruebas, en la terminal se debe correr el comando:

**vendor/bin/phpunit tests/UsuariosTest.php**



```
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.  
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/UsuariosTest.php
```

Este comando ejecuta todas las pruebas dentro del directorio /tests y muestra un reporte con resultados.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE  
  
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.  
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/UsuariosTest.php  
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.  
  
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.  
  
Runtime: PHP 8.2.12  
  
..... 5 / 5 (100%)  
..... 5 / 5 (100%)  
  
*****  
Todas las pruebas del módulo USUARIOS se realizaron correctamente ✓  
Se validaron agregar, consultar, actualizar , así como consultar roles, estados y géneros.  
*****  
  
Time: 00:00.120, Memory: 8.00 MB  
  
OK (5 tests, 13 assertions)  
PS C:\xampp\htdocs\Bootstrap\Bootstrap> |
```





## Documentación de Pruebas del Módulo de Ticket de Ingreso con PHPUnit

### 1. Explicación de las Pruebas del Módulo Ticket de Ingreso (Método por Método)

#### 2.1. testAgregarIngreso()

- **Qué prueba:**  
Verifica que el método AgregarIngreso() pueda insertar un ingreso de varios productos sin lanzar excepciones.
- **Cómo:**  
Se llama a AgregarIngreso() con datos de prueba válidos.  
No se espera un valor de retorno, por eso la prueba es que no lance excepción.

```
Windsurf: Refactor | Explain
class IngresoTest extends TestCase
{
    protected $ingreso;
    protected $producto;

    protected static $existingProductId = 301; // ID de un producto que ya existe en tu tabla 'producto'
    protected static $existingEmployeeDocument = '1080180837'; // Num_Documento de un usuario/empleador que ya existe en tu tabla 'usuario'

    Windsurf: Refactor | Explain | Generate Function Comment | X
    protected function setUp(): void
    {
        $this->ingreso = new Ingreso();
        $this->producto = new Producto();
    }
}
```

```
Windsurf: Refactor | Explain | Generate Function Comment | X
public function testAgregarIngreso()
{
    $productoCodigoProducto = self::$existingProductId;
    $empleador = self::$existingEmployeeDocument;

    $fechaIngreso = date('Y-m-d');
    $cantidad = 5;
    $descripcion = 'Ingreso de prueba exitoso';
    $precioIngreso = 25000;
    $productoExistente = $this->producto->consultarProducto($productoCodigoProducto)->fetch_assoc();
    $this->assertNotNull($productoExistente, "El producto con ID " . $productoCodigoProducto . " debe existir para ejecutar esta prueba.");

    $stockInicial = $productoExistente['Stock'];
    try {
        $this->ingreso->agregarIngreso(
            $fechaIngreso,
            $productoCodigoProducto,
            $cantidad,
            $descripcion,
            $precioIngreso,
            $empleador
        );
        $this->assertTrue(true, 'El ingreso debería agregarse correctamente y el stock actualizarse.');
```

```
        // Verificar que el stock se actualizó correctamente
        $productoFinal = $this->producto->consultarProducto($productoCodigoProducto)->fetch_assoc();
        $this->assertEquals($stockInicial + $cantidad, $productoFinal['Stock'], 'El stock del producto debería haberse incrementado.');
```

```
    } catch (\Exception $e) {
        $this->fail('El método agregarIngreso lanzó una excepción: ' . $e->getMessage());
    }
}
```

Servidor: 127.0.0.1
Base de datos: paty\_sport
Tabla: detalle\_ingreso

[Examinar](#)
[Estructura](#)
[SQL](#)
[Buscar](#)
[Insertar](#)
[Exportar](#)
[Importar](#)
[Privilegios](#)
[Operaciones](#)
[Disparadores](#)

Mostrando filas 0 - 15 (total de 16, La consulta tardó 0,0002 segundos.)

SELECT \* FROM `detalle\_ingreso`

☐ Perfilando
 [\[ Editar en línea \]](#)
[\[ Editar \]](#)
[\[ Explicar SQL \]](#)
[\[ Crear código PHP \]](#)
[\[ Actualizar \]](#)

☐ Mostrar todo
 Número de filas: 
Filtrar filas: 
Ordenar según la clave:

Opciones extra

			ProductoCodigoProducto	TicketSalidaTicketingreso	Cantidad	Descripción	PrecioIngreso	Empleado
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	301	1	6 NULL	17000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	301	8	40 0	17000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	301	19	5 0	25000	1080180837
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	302	5	8 NULL	25000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	303	4	8 NULL	32000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	304	3	10 NULL	35000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	304	11	3 0	34000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	306	2	10 NULL	24000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	308	17	4 0	40000	190831983
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	310	9	25 0	500	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	310	10	3 0	500	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	317	12	20 0	50000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	318	13	17 0	70000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	319	14	3 0	50000	52747225
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	324	16	15 0	45000	1080180837
<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	330	18	4 0	3000	9346500

- **Validación:**  
Si no se lanza ninguna excepción, la prueba pasa.



```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK POSTMAN CONSOLE

PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/IngresoTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

..                                                    2 / 2 (100%)

*****
Todas las pruebas del módulo INGRESOS se realizaron correctamente ✓
Se validaron agregar y consultar ingresos.
*****

Time: 00:00.090, Memory: 8.00 MB

OK (2 tests, 6 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```

2.2. testConsultarTodosLosIngresos()

- **Qué prueba:**  
Verifica que el método ConsultarTodosLosIngresos() me consulte correctamente los datos de todos los ticket de ingreso existentes.
- **Cómo:**  
Se llama a ConsultarTodosLosIngresos() porque obtiene todos los ingresos que hacen de los productos y los muestra.

```
Windsurf: Refactor | Explain | Generate Function Comment | X
public function testConsultarTodosLosIngresos()
{
    $resultado = $this->ingreso->consultarIngreso(); // Sin parámetro consulta todos

    $this->assertInstanceOf(mysqli_result::class, $resultado, 'consultarIngreso sin parámetro debería devolver un objeto mysqli_result');
    $this->assertGreaterThanOrEqual(0, $resultado->num_rows, 'Debería devolver al menos cero ingresos');

    $resultado->free();
}
```

Entrada de Productos

Estas aquí:  
[Ingreso](#)

Agregar Entrada

INGRESOS DE PRODUCTOS

No° Entrada	Fecha de Ingreso	Producto	11 Cantidad	Descripción	\$ Valor de Ingreso	Precio Total	Personal
12	2024-11-18	Camisa polo	20	0	\$50.000	\$1.000.000	Viviana
13	2024-11-18	Jean Panaa	17	0	\$70.000	\$1.190.000	Viviana
14	2024-11-19	drjrd	3	0	\$50.000	\$150.000	Viviana
16	2025-02-11	camisa	15	0	\$45.000	\$675.000	Lilley
17	2025-04-08	Buzo Overzice	4	0	\$40.000	\$160.000	Andres
18	2025-04-08	Pelusa	4	0	\$3.000	\$12.000	Javier
19	2025-05-20	Sudadera Jonny y Nayri	5	0	\$25.000	\$125.000	Lilley
20	2025-05-20	Sudadera Jonny y Nayri	5	0	\$25.000	\$125.000	Lilley
21	2025-05-20	Sudadera Jonny y Nayri	5	0	\$25.000	\$125.000	Lilley
22	2025-05-20	Sudadera Jonny y Nayri	5	0	\$25.000	\$125.000	Lilley
23	2025-05-20	Sudadera Jonny y Nayri	5	0	\$25.000	\$125.000	Lilley



- **Validación:**

El método debe retornar true, indicando que la actualización fue exitosa.

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/IngresoTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

..                                                    2 / 2 (100%)

*****
Todas las pruebas del módulo INGRESOS se realizaron correctamente ✓
Se validaron agregar y consultar ingresos.
*****

Time: 00:00.090, Memory: 8.00 MB

OK (2 tests, 6 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```

### 3. Consideraciones Adicionales

- **Dependencias de Datos:**

Para que las pruebas funcionen correctamente, los productos a darle ingreso deben ser existentes en sus respectivas tablas de la base de datos.

- **Configuración de Base de Datos:**

Se recomienda tener una base de datos de pruebas para no afectar los datos reales del proyecto.

- **Actualización de Stock en la Tabla Productos:**

Se hace la debida inserción del stock en agregar ingreso para que ella lleve esa cantidad a la tabla de productos.

```
// Verificar que el stock se actualizó correctamente
$productoFinal = $this->producto->consultarProducto($productoCodigoProducto)->fetch_assoc();
$this->assertEquals($stockInicial + $cantidad, $productoFinal['Stock'], 'El stock del producto debería haberse incrementado.');
```

- **Método tearDownAfterClass():**

Al finalizar todas las pruebas, imprime un mensaje en consola indicando que todas las pruebas del módulo de ticket de ingreso se realizaron con éxito.



```
Windsurf: Refactor | Explain | Generate Function Comment | X
public static function tearDownAfterClass(): void
{
    echo "\n\n*****\n";
    echo "Todas las pruebas del módulo INGRESOS se realizaron correctamente ✅\n";
    echo "Se validaron agregar y consultar ingresos.\n";
    echo "*****\n\n";
}
}
```

4. Ejecución y Resultado de las Pruebas

Al ejecutar vendor/bin/phpunit en la raíz del proyecto, se corren automáticamente todas las pruebas definidas en IngresoTest.php. Si todas las pruebas pasan, el resultado será algo similar a:

```
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit
tests/IngresoTest.php
```

PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.12

.. 2 / 2 (100%)

\*\*\*\*\*

Todas las pruebas del módulo INGRESOS se realizaron correctamente ✅

Se validaron agregar y consultar ingresos.

\*\*\*\*\*

Time: 00:00.298, Memory: 8.00 MB

OK (2 tests, 6 assertions)

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/IngresoTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.12

.. 2 / 2 (100%)

*****
Todas las pruebas del módulo INGRESOS se realizaron correctamente ✅
Se validaron agregar y consultar ingresos.
*****

Time: 00:00.298, Memory: 8.00 MB

OK (2 tests, 6 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```



## 5. Ejecución de las Pruebas

Para ejecutar las pruebas, en la terminal se debe correr el comando:

**vendor/bin/phpunit tests/IngresoTest.php**

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/IngresoTest.php
```

Este comando ejecuta todas las pruebas dentro del directorio /tests y muestra un reporte con resultados.

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/IngresoTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.2.12

..                                                     2 / 2 (100%)

*****
Todas las pruebas del módulo INGRESOS se realizaron correctamente ✓
Se validaron agregar y consultar ingresos.
*****

Time: 00:00.298, Memory: 8.00 MB

OK (2 tests, 6 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```



## Documentación de Pruebas del Módulo de Ticket de Salida con PHPUnit

### 1. Explicación de las Pruebas del Módulo Ticket de Salida (Método por Método)

#### 2.1. testAgregarSalidaExitosa()

- **Qué prueba:**  
Verifica que el método AgregarSalidaExitosa() puede insertar una salida de varios productos sin lanzar excepciones.
- **Cómo:**  
Se llama a AgregarSalidaExitosa() con datos de prueba válidos.  
No se espera un valor de retorno, por eso la prueba es que no lance excepción.
- 

```
WINDSON: REFACTOR | Explain
class SalidaTest extends TestCase
{
    protected $salida;
    protected $producto;

    protected static $existingProductId = 301;
    protected static $existingEmployeeDocument = '1080180837';
}
```

```
public function testAgregarSalidaExitosa()
{
    $productoCodigo = self::$existingProductId;
    $empleado = self::$existingEmployeeDocument;
    $fechaSalida = date('Y-m-d');
    $cantidad = 2;
    $productoExistente = $this->producto->consultarProducto($productoCodigo)->fetch_assoc();
    $this->assertNotNull($productoExistente, "El producto con ID " . $productoCodigo . " debe existir para ejecutar esta prueba.");
    $stockInicial = $productoExistente['Stock'];
    if ($stockInicial < $cantidad) {
        $nuevoStockParaTest = $stockInicial + $cantidad + 5;
        $conexion = Conectarse();
        $sqlUpdateStock = "UPDATE producto SET Stock = ? WHERE CodigoProducto = ?";
        $stmtUpdateStock = $conexion->prepare($sqlUpdateStock);
        $stmtUpdateStock->bind_param("ii", $nuevoStockParaTest, $productoCodigo);
        $stmtUpdateStock->execute();
        $stmtUpdateStock->close();
        $conexion->close();
        $productoExistente = $this->producto->consultarProducto($productoCodigo)->fetch_assoc();
        $stockInicial = $productoExistente['Stock'];
    }

    try {
        $this->salida->agregarSalida(
            $fechaSalida,
            $cantidad,
            $productoCodigo,
            $empleado
        );
        $this->assertTrue(true, 'La salida debería agregarse correctamente y el stock actualizarse.');
```



Servidor: 127.0.0.1 » Base de datos: paty\_sport » Tabla: ticket\_ingreso

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Disparadores

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

				IdTicketingreso	FechaIngresoProducto	PrecioTotal
<input type="checkbox"/>	Editar	Copiar	Borrar	1	2024-07-13	102000
<input type="checkbox"/>	Editar	Copiar	Borrar	2	2024-07-17	240000
<input type="checkbox"/>	Editar	Copiar	Borrar	3	2024-07-17	35000
<input type="checkbox"/>	Editar	Copiar	Borrar	4	2024-07-31	192000
<input type="checkbox"/>	Editar	Copiar	Borrar	5	2024-08-07	200000
<input type="checkbox"/>	Editar	Copiar	Borrar	8	2024-10-16	680000
<input type="checkbox"/>	Editar	Copiar	Borrar	9	2024-10-16	12500
<input type="checkbox"/>	Editar	Copiar	Borrar	10	2024-10-16	1500
<input type="checkbox"/>	Editar	Copiar	Borrar	11	2024-11-18	102000
<input type="checkbox"/>	Editar	Copiar	Borrar	12	2024-11-18	1000000
<input type="checkbox"/>	Editar	Copiar	Borrar	13	2024-11-18	1190000
<input type="checkbox"/>	Editar	Copiar	Borrar	14	2024-11-19	150000
<input type="checkbox"/>	Editar	Copiar	Borrar	16	2025-02-11	675000
<input type="checkbox"/>	Editar	Copiar	Borrar	17	2025-04-08	160000
<input type="checkbox"/>	Editar	Copiar	Borrar	18	2025-04-08	12000
<input type="checkbox"/>	Editar	Copiar	Borrar	19	2025-05-20	125000
<input type="checkbox"/>	Editar	Copiar	Borrar	20	2025-05-20	125000
<input type="checkbox"/>	Editar	Copiar	Borrar	21	2025-05-20	125000
<input type="checkbox"/>	Editar	Copiar	Borrar	22	2025-05-20	125000
<input type="checkbox"/>	Editar	Copiar	Borrar	23	2025-05-20	125000
<input type="checkbox"/>	Editar	Copiar	Borrar	24	2025-05-20	125000
<input type="checkbox"/>	Editar	Copiar	Borrar	25	2025-05-20	125000
<input type="checkbox"/>	Editar	Copiar	Borrar	26	2025-05-20	125000

Servidor: 127.0.0.1 » Base de datos: paty\_sport » Tabla: detalle\_salida

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Disparadores

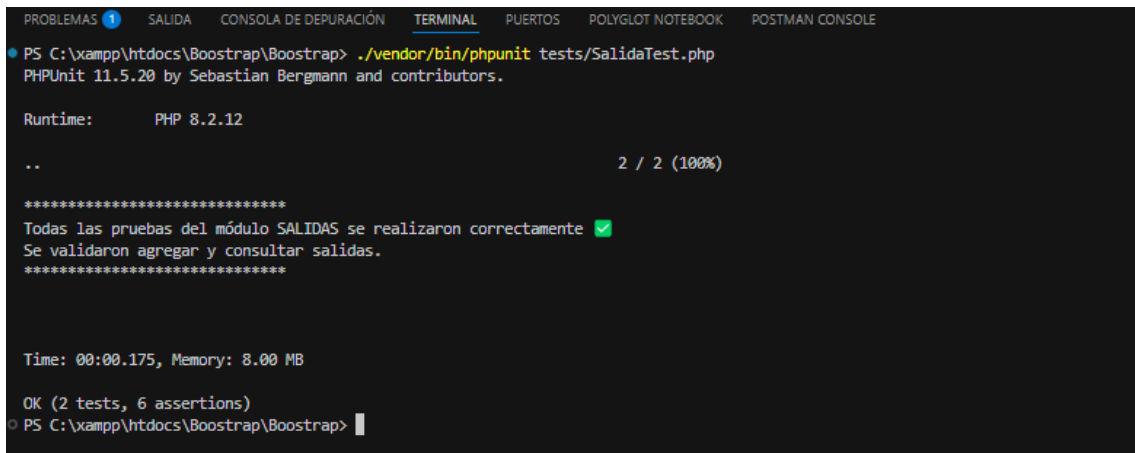
☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

					ProductoCodigoProducto	SalidaIdTicketSalida	Cantidad	ValorUnitario	Empleado
<input type="checkbox"/>	Editar	Copiar	Borrar		301	15	2	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		301	16	1	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		301	17	1	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		301	18	2	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		301	19	1	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		301	20	2	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		301	21	1	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		301	22	2	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		301	23	2	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		301	24	2	40000	1080180837
<input type="checkbox"/>	Editar	Copiar	Borrar		302	3	3	60000	80461462
<input type="checkbox"/>	Editar	Copiar	Borrar		302	5	3	60000	80461462
<input type="checkbox"/>	Editar	Copiar	Borrar		302	7	1	60000	1033743193
<input type="checkbox"/>	Editar	Copiar	Borrar		303	6	2	30000	1033743193
<input type="checkbox"/>	Editar	Copiar	Borrar		307	8	2	30000	80461462
<input type="checkbox"/>	Editar	Copiar	Borrar		309	10	2	30000	93481465
<input type="checkbox"/>	Editar	Copiar	Borrar		315	2	3	40000	80461462
<input type="checkbox"/>	Editar	Copiar	Borrar		319	4	1	30000	80461462
<input type="checkbox"/>	Editar	Copiar	Borrar		324	9	10	45000	80461462

- **Validación:**  
Si no se lanza ninguna excepción, la prueba pasa.





- **Qué prueba:**

- **Cómo:**

Salida de Productos

Estos aquí:  
Salidas

Agregar Salida

SALIDAS DE PRODUCTOS

No° Salida	Fecha de Salida	Producto	11 Cantidad	\$ Valor Unitario	Valor Total	Personal
15	2025-05-20	Sudadera Jonny y Nayri	2	\$40.000	\$95.200	Lilley
16	2025-05-20	Sudadera Jonny y Nayri	1	\$40.000	\$47.600	Lilley
17	2025-05-20	Sudadera Jonny y Nayri	1	\$40.000	\$47.600	Lilley
18	2025-05-20	Sudadera Jonny y Nayri	2	\$40.000	\$95.200	Lilley
19	2025-05-20	Sudadera Jonny y Nayri	1	\$40.000	\$47.600	Lilley
20	2025-05-20	Sudadera Jonny y Nayri	2	\$40.000	\$95.200	Lilley
21	2025-05-20	Sudadera Jonny y Nayri	1	\$40.000	\$47.600	Lilley
22	2025-05-20	Sudadera Jonny y Nayri	2	\$40.000	\$95.200	Lilley
23	2025-05-20	Sudadera Jonny y Nayri	2	\$40.000	\$95.200	Lilley
24	2025-05-20	Sudadera Jonny y Nayri	2	\$40.000	\$95.200	Lilley
25	2025-05-20	Sudadera Jonny y Nayri	2	\$40.000	\$95.200	Lilley

59:00

El método debe retornar true, indicando que la actualización fue exitosa.



```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/SalidaTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.2.12

..                                                    2 / 2 (100%)

*****
Todas las pruebas del módulo SALIDAS se realizaron correctamente ✓
Se validaron agregar y consultar salidas.
*****

Time: 00:00.175, Memory: 8.00 MB

OK (2 tests, 6 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```

### 3. Consideraciones Adicionales

- **Dependencias de Datos:**  
Para que las pruebas funcionen correctamente, los productos a darle salida deben ser existentes en sus respectivas tablas de la base de datos.
- **Configuración de Base de Datos:**  
Se recomienda tener una base de datos de pruebas para no afectar los datos reales del proyecto.
- **Actualización de Stock en la Tabla Productos:**  
Se hace la debida inserción del stock en agregar una salida para que ella lleve esa cantidad a la tabla de productos.

```
// Verificar que el stock se actualizó correctamente
$productoFinal = $this->producto->consultarProducto($productoCodigo)->fetch_assoc();
$this->assertEquals($stockInicial - $cantidad, $productoFinal['Stock'], 'El stock del producto debería haberse decrementado.');
```

- **Método tearDownAfterClass():**  
Al finalizar todas las pruebas, imprime un mensaje en consola indicando que todas las pruebas del módulo de ticket de salida se realizaron con éxito.

```
public static function tearDownAfterClass(): void
{
    echo "\n\n*****\n\n";
    echo "Todas las pruebas del módulo SALIDAS se realizaron correctamente ✓\n\n";
    echo "Se validaron agregar y consultar salidas.\n\n";
    echo "*****\n\n";
}
```



#### 4. Ejecución y Resultado de las Pruebas

Al ejecutar `vendor/bin/phpunit` en la raíz del proyecto, se corren automáticamente todas las pruebas definidas en `SalidaTest.php`. Si todas las pruebas pasan, el resultado será algo similar a:

```
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/SalidaTest.php
```

PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.12

Runtime: PHP 8.2.12

.. 2 / 2 (100%)

\*\*\*\*\*

Todas las pruebas del módulo SALIDAS se realizaron correctamente ✓

Se validaron agregar y consultar salidas.

\*\*\*\*\*

Time: 00:00.175, Memory: 8.00 MB

OK (2 tests, 6 assertions)

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/SalidaTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.12

.. 2 / 2 (100%)

*****
Todas las pruebas del módulo SALIDAS se realizaron correctamente ✓
Se validaron agregar y consultar salidas.
*****

Time: 00:00.175, Memory: 8.00 MB

OK (2 tests, 6 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap> |
```

#### 5. Ejecución de las Pruebas

Para ejecutar las pruebas, en la terminal se debe correr el comando:

**`vendor/bin/phpunit tests/IngresoTest.php`**

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/SalidaTest.php
```



Este comando ejecuta todas las pruebas dentro del directorio /tests y muestra un reporte con resultados.

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK POSTMAN CONSOLE
PS C:\xampp\htdocs\Bootstrap\Bootstrap> ./vendor/bin/phpunit tests/SalidaTest.php
PHPUnit 11.5.20 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.2.12

..                                                     2 / 2 (100%)

*****
Todas las pruebas del módulo SALIDAS se realizaron correctamente ✓
Se validaron agregar y consultar salidas.
*****

Time: 00:00.175, Memory: 8.00 MB

OK (2 tests, 6 assertions)
PS C:\xampp\htdocs\Bootstrap\Bootstrap>
```

## Conclusión

La integración de PHPUnit en el proyecto PHP puro permitió asegurar la calidad y estabilidad del módulo de productos, usuarios, ticket de salida y ticket de ingreso. Las pruebas unitarias desarrolladas cubren las operaciones CRUD principales y las consultas a tablas relacionadas, facilitando la detección temprana de errores y regresiones en el código.