



**Servicio Nacional de Aprendizaje SENA**

**Centro Electricidad, Electrónica y  
Telecomunicaciones**



## **PRUEBAS UNITARIAS**

### **SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE CLIENTES Y COTIZACIONES DE SEGUROS**

**SENA 2022**



**Servicio Nacional de Aprendizaje SENA**

**Centro Electricidad, Electrónica y  
Telecomunicaciones**



## **ÍNDICE**

<b>ÍNDICE</b>	<b>2</b>
<b>Testing Cliente.</b>	<b>3</b>
<b>Testing Póliza</b>	<b>4</b>
<b>Testing Siniestro.</b>	<b>6</b>
<b>Testing Usuario.</b>	<b>8</b>
<b>Testing Search.</b>	<b>10</b>
<b>Testing Connection.</b>	<b>12</b>



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



## 1. Testing Cliente.

- Crear, modificar y eliminar un cliente.

Error en tests update problema al instanciar la variable \$documento2.

```
Testing_cliente.php x
28 $nombre2='David';
29 $apellido1='Lopez';
30 $apellido2='Leal';
31 $fecha='2022-06-02';
32 $correo='lopez@gmail.com';
33 $cel='311000';
34 $cel2='315000';
35 $direc1='c112';
36 $direc2='c113';
37 $tdocum=2;
38 $ciudad=5;
39 $test=((new Cliente($correo,$direc1,$direc2,$cel,$cel2,$nombre1,$nombre2,$apellido1,$apellido2,
40 $tdocum,$fecha,$ciudad)) ->update($documento));
41 $this->assertEquals( expected: 1,$test);

Terminal: Local x
1) Testing_cliente::test_update
Error: Cannot access uninitialized non-nullable property Models\Cliente::$documento by reference

C:\xampp\htdocs\Perez Lara\Models\Cliente.php:47
C:\xampp\htdocs\Perez Lara\Models\Cliente.php:92
C:\xampp\htdocs\Perez Lara\Test\Testing_cliente.php:40

ERRORS!
Tests: 3, Assertions: 2, Errors: 1.
PS C:\xampp\htdocs\Perez Lara> Z
```

Error de referencia de variable “\$documento2” corregido

```
Testing_cliente.php x
1 <?php
2 use Models\Cliente;
3
4 class Testing_cliente extends \PHPUnit\Framework\TestCase {
5
6     public function test_create(){
7         $documento='5151';
8         $nombre1='oscar';
9         $nombre2='Giovany';
10        $apellido1='Lopez';
11        $apellido2='Leal';
12        $fecha='2022-06-02';
13        $correo='lopez@gmail.com';
14        $cel='311000';
15        $cel2='315000';
16        $direc1='c112';
17        $direc2='c113';
18        $tdocum=2;
19        $ciudad=5;
20        $test=((new Cliente($documento,$correo,$direc1,$direc2,$cel,$cel2,$nombre1,$nombre2,$apellido1,$apellido2,$tdocum,
21        $fecha,$ciudad)) ->create());
22        $this->assertEquals( expected: 1,$test);
23    }
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



```
public function test_update(){
    $documento='5151';
    $documento2='5151';
    $nombre1='Juan';
    $nombre2='David';
    $apellido1='Lopez';
    $apellido2='Leal';
    $fecha='2022-06-02';
    $correo='lopez@gmail.com';
    $cel='311000';
    $cel2='315000';
    $direc1='c112';
    $direc2='c113';
    $tdocum=2;
    $ciudad=5;
    $test=((new Cliente($documento,$correo,$direc1,$direc2,$cel,$cel2,$nombre1,$nombre2,$apellido1,$apellido2,
    $tdocum,$fecha,$ciudad))→update($documento2));
    $this→assertEquals( expected: 1,$test);
}

public function test_delete()
{
    $documento = '5151';

    $test = ((new Cliente($documento))→delete());
    $this→assertEquals( expected: 1, $test);
}
```

Prueba a la clase Cliente, donde se ejecutan tres funciones (create, update y delete) diferentes para validar el modelo Cliente.

Testing\_cliente.php

```
<?php
use Models\Cliente;

class Testing_cliente extends \PHPUnit\Framework\TestCase {

    public function test_create(){
        $documento='5151';
        $nombre1='oscar';
        $nombre2='Siswany';
        $apellido1='Lopez';
        $apellido2='Leal';
        $fecha='2022-06-02';
        $correo='lopez@gmail.com';
        $cel='311000';
        $cel2='315000';
        $direc1='c112';
        $direc2='c113';
        $tdocum=2;
        $ciudad=5;
        $test=((new Cliente($documento,$correo,$direc1,$direc2,$cel,$cel2,$nombre1,$nombre2,$apellido1,$apellido2,$tdocum,
        $fecha,$ciudad))→create());
        $this→assertEquals( expected: 1,$test);
    }

    public function test_update(){
        $documento='5152';
        $documento2='5151';
        $nombre1='oscar';
        $nombre2='David';
        $apellido1='Lopez';
        $apellido2='Leal';
        $fecha='2022-06-02';
        $correo='lopez@gmail.com';
        $cel='311000';
        $cel2='315000';
        $direc1='c112';
        $direc2='c113';
        $tdocum=2;
        $ciudad=5;
        $test=((new Cliente($documento,$correo,$direc1,$direc2,$cel,$cel2,$nombre1,$nombre2,$apellido1,$apellido2,
        $tdocum,$fecha,$ciudad))→update($documento2));
        $this→assertEquals( expected: 1,$test);
    }

    public function test_delete()
    {
        $documento = '5151';

        $test = ((new Cliente($documento))→delete());
        $this→assertEquals( expected: 1, $test);
    }
}
```

Terminal: Local

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/powershell>

PS C:\xampp\htdocs\Perez Lara> .\vendor\bin\phpunit Test/Testing\_cliente.php
PHPUnit 9.5.23 #StandWithUkraine

...
Time: 00:00.095, Memory: 4.00 MB

OK (3 tests, 3 assertions)
PS C:\xampp\htdocs\Perez Lara>



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



## 2. Testing Póliza

Crear y modificar una póliza.

```
Testing_poliza.php x
1 <?php
2 use Models\Poliza;
3 use Models\Cliente;
4 class Testing_poliza extends \PHPUnit\Framework\TestCase {
5
6     public function test_create(){
7         $codigo=45;
8         $docuCli = (new \Models\Cliente('1010760139'));
9         $valorAsegurado=10000;
10        $valorPrima=12258;
11        $seguro=5;
12        $aseguradora=4;
13        $archivo='archivo.pdf';
14        $estado=3;
15        $test=((new Poliza($codigo,$docuCli,$valorAsegurado,$valorPrima,$seguro,$aseguradora,$archivo,$estado))
16            →create());
17        $this→assertEquals( expected: 1,$test);
18    }
19
20    public function test_Update(){
21        $codigo=45;
22        $codigo2=46;
23        $docuCli = (new \Models\Cliente('1010760139'));
24        $valorAsegurado=10000;
25        $valorPrima=12258;
26        $seguro=5;
27        $aseguradora=4;
28        $archivo='archivo.pdf';
29        $estado=4;
30
31        $test=((new Poliza($codigo2,$docuCli,$valorAsegurado,$valorPrima,$seguro,$aseguradora,$archivo,$estado))
32            →update($codigo));
33        $this→assertEquals( expected: 1,$test);
34    }
35
36
37 }
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



Prueba a la clase Póliza, donde se ejecutan dos funciones (create y update) diferentes para validar el modelo Poliza.

```
Testing_poliza.php
<?php
use Models\Poliza;
use Models\Cliente;

class Testing_poliza extends \PHPUnit\Framework\TestCase {

    public function test_create(){
        $codigo=45;
        $docuCli = (new \Models\Cliente('1010760139'));
        $valorAsegurado=10000;
        $valorPrima=12258;
        $seguro=5;
        $aseguradora=4;
        $archivo= 'archivo.pdf';
        $estado=3;
        $test=((new Poliza($codigo,$docuCli,$valorAsegurado,$valorPrima,$seguro,$aseguradora,$archivo,$estado))
        ->create());
        $this->assertEquals( expected: 1,$test);
    }

    public function test_update(){
        $codigo=45;
        $codigo2=46;
        $docuCli = (new \Models\Cliente('1010760139'));
        $valorAsegurado=10000;
        $valorPrima=12258;
        $seguro=5;
        $aseguradora=4;
        $archivo= 'archivo.pdf';
        $estado=4;

        $test=((new Poliza($codigo2,$docuCli,$valorAsegurado,$valorPrima,$seguro,$aseguradora,$archivo,$estado))
        ->update($codigo));
        $this->assertEquals( expected: 1,$test);
    }
}
```

```
Terminal: Local x + v
PS C:\xampp\htdocs\Perez Lara> ./vendor/bin/phpunit Test/Testing_poliza.php
PHPUnit 9.5.23 #StandWithUkraine

..
2 / 2 (100%)

Time: 00:00.027, Memory: 4.00 MB

OK (2 tests, 2 assertions)
PS C:\xampp\htdocs\Perez Lara>
```



### 3. Testing Siniestro.

- Crear y modificar un siniestro.

```
Testing_siniestro.php x
1  <?php
2
3  use ...
4
5
6  class Testing_siniestro extends \PHPUnit\Framework\TestCase
7  {
8
9      public function test_Create()
10     {
11         $codigo = (new \Models\Poliza(13));
12         $descripcion = 'Testo ejemplo';
13         $monto = '15200';
14         $anexo = "archivo.pdf";
15         $estado = 2;
16
17         $test = ((new Siniestro($descripcion,$monto,$codigo,$anexo,$estado))>create());
18         $this->assertEquals( expected: 1, $test);
19     }
20     public function test_Update()
21     {
22         $id = 8;
23         $codigo = (new \Models\Poliza(13));
24         $descripcion = 'actualizada';
25         $monto = '15200';
26         $anexo = "archivo.pdf";
27         $estado = 2;
28
29         $test = ((new Siniestro($id,$descripcion,$monto,$codigo,$anexo,$estado))>update());
30         $this->assertEquals( expected: 1, $test);
31     }
32 }
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



Prueba a la clase Siniestro, donde se ejecutan dos funciones (create y update) diferentes para validar el modelo Siniestro.

```
Testing_siniestro.php
1 <?php
2
3 use PHPUnit\Framework\TestCase;
4
5 class Testing_siniestro extends \PHPUnit\Framework\TestCase
6 {
7
8     public function test_Create()
9     {
10         $codigo = (new \Models\Poliza(455));
11         $descripcion = 'Testo ejemplo';
12         $monto = '15200';
13         $anexo = "archivo.pdf";
14         $estado = 2;
15
16         $test = ((new Siniestro($descripcion,$monto,$codigo,$anexo,$estado))>create());
17         $this->assertEquals(1, $test);
18     }
19
20     public function test_Update()
21     {
22         $id = 8;
23         $codigo = (new \Models\Poliza(455));
24         $descripcion = 'actualizada';
25         $monto = '15200';
26         $anexo = "archivo.pdf";
27         $estado = 2;
28
29         $test = ((new Siniestro($id,$descripcion,$monto,$codigo,$anexo,$estado))>update());
30         $this->assertEquals(1, $test);
31     }
32 }
```

```
Terminal: Local
PS C:\xampp\htdocs\Perez Lara> ./vendor/bin/phpunit Test/Testing_siniestro.php
PHPUnit 9.5.23 #StandWithUkraine

.F

Time: 00:00.027, Memory: 4.00 MB

There was 1 failure:

1) Testing_siniestro::test_Update
Failed asserting that 0 matches expected 1.

C:\xampp\htdocs\Perez Lara\Test\Testing_siniestro.php:30

FAILURES!
Tests: 2, Assertions: 2, Failures: 1.
PS C:\xampp\htdocs\Perez Lara>
```

## 4. Testing Usuario.

- Crear, modificar un usuario y validación del login.

primera prueba error en la función create error valor duplicado

```
Testing_usuario.php
1 public function test_update()
2 {
3     $nombre1='juan';
4     $nombre2='felipe';
5     $apellido1='diaz';
6     $apellido2='Leal';
7     $documento='51511';
8     $documento2='00000';
9     $correo='lopez@gmail.com';
10    $cel='311000';
11    $Idocum=2;
12    $foto='foto_usuario';
13    $clave='clave_usuario';
14 }
```

```
Terminal: Local
1) Testing_usuario::test_create
PDOException: SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry
'lopez@gmail.com' for key 'email'

C:\xampp\htdocs\Perez Lara\Models\Usuario.php:72
C:\xampp\htdocs\Perez Lara\Test\Testing_usuario.php:21

ERRORS!
Tests: 3, Assertions: 2, Errors: 1, Failures: 1.
PS C:\xampp\htdocs\Perez Lara>
```





Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



## Error de entrada duplicada corregido

```
Testing_usuario.php X
1  <?php
2
3  use Models\Poliza;
4  use Models\Usuario;
5
6  class Testing_usuario extends \PHPUnit\Framework\TestCase
7  {
8
9      public function test_create()
10     {
11         $nombre1='oscar';
12         $nombre2='Giovany';
13         $apellido1='Lopez';
14         $apellido2='Leal';
15         $documento1='51511';
16         $correo='lopez@gmail.com';
17         $cel='311000';
18         $Idocum=2;
19         $foto='foto_usuario';
20         $clave='clave_usuario';
21         $test = ((new Usuario($documento1,$nombre1,$nombre2,$apellido1,$apellido2,$cel,$correo,$clave,$foto,$Idocum))->create());
22         $this->assertEquals( expected: 1, $test);
23     }
24
25     public function test_update()
26     {
27         $nombre1='juan';
28         $nombre2='felipe';
29         $apellido1='diaz';
30         $apellido2='Leal';
31         $documento='51511';
32         $documento2='00000';
33         $correo='lopez@gmail.com';
34         $cel='311000';
35         $Idocum=2;
36         $foto='foto_usuario';
37         $clave='clave_usuario';
38
39         $test = ((new Usuario($documento2, $nombre1, $nombre2, $apellido1, $apellido2,$cel,$correo,$clave,$foto,$Idocum))->update($documento));
40         $this->assertEquals( expected: 1, $test);
41     }
42
43     public function test_Validation_Login(){
44         $test = ((new Usuario('admin@perezlara.com', 'adminperezlara'))->getCredentials());
45         $this->assertEquals( expected: true, $test);
46     }
47 }
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



Prueba a la clase Usuario, donde se ejecutan tres funciones (create, update y validar) diferentes para validar el modelo Usuario.

```
Testing_usuario.php x
22 $this->assertEquals( expected: 1, $test);
23 }
24
25 public function test_update()
26 {
27     $nombre1='juan';
28     $nombre2='felipe';
29     $apellido1='diaz';
30     $apellido2='Leal';
31     $documento='51511';
32     $documento2='00000';
33     $correo='lopez@gmail.com';
34     $cel='311000';
35     $idocum=2;
36     $foto='foto_usuario';
37     $clave='clave_usuario';
38
39     $test = ((new Usuario($documento2, $nombre1, $nombre2, $apellido1, $apellido2, $cel, $correo, $clave, $foto, $idocum))->update($documento));
40     $this->assertEquals( expected: 1, $test);
41 }
42
43
44 public function test_Validation_Login(){
45     $test = ((new Usuario('admin@perezlara.com', 'adminperezlara'))->getCredenciales());
46     $this->assertEquals( expected: true, $test);
47 }
48
Testing_usuario test_update()
```

```
Terminal: Local x + e
PS C:\xampp\htdocs\Perez Lara> ./vendor/bin/phpunit Test/Testing_usuario.php
PHPUnit 9.5.24 #StandWithUkraine

...
3 / 3 (100%)

Time: 00:00.111, Memory: 4.00 MB

OK (3 tests, 3 assertions)
PS C:\xampp\htdocs\Perez Lara>
```

## 5. Testing Search.

- Validar búsquedas o consultas de base de datos.

```
Testing_search.php x
1 <?php
2
3 use Models\Cliente;
4
5 class Testing_search extends \PHPUnit\Framework\TestCase{
6
7     public function test_search()
8     {
9
10         $consulta='1010';
11         $test = ((new Cliente())->searchClient( consulta: '%'.$consulta.'%'));
12         $this->assertIsArray($test, message: "array");
13     }
14
15 }
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



Prueba a la clase Cliente, donde se ejecuta la función (searchClient) para validar el modelo Cliente.

```
Testing_search.php x
1  <?php
2
3  use Models\Cliente;
4
5  class Testing_search extends \PHPUnit\Framework\TestCase{
6
7  public function test_search()
8  {
9
10     $consulta='1010';
11     $test = ((new Cliente())->searchClient( consulta: '%'.$consulta.'%'));
12     $this->assertIsArray($test, message: "array");
13 }
14
15 }
16
```

```
Terminal: Local x +
PS C:\xampp\htdocs\Perez Lara> ./vendor/bin/phpunit Test/Testing_search.php
PHPUnit 9.5.24 #StandWithUkraine

.

1 / 1 (100%)

Time: 00:00.011, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\xampp\htdocs\Perez Lara>
```

## 6. Testing Connection.

- Validar conexión a base de datos.

Error en el retorno de conexión base de datos

```
Testing_connection.php x Connection.php x
1  <?php
2  use Test\Connection;
3  require ('Connection.php');
4
5  class Testing_connection extends \PHPUnit\Framework\TestCase
6
7
8  public function test_conec()
9  {
10     $conec=(new Connection()) -> connect();
11     $this->assertIsObject($conec, message: "Es de tipo objeto o no");
12 }
13
```

```
Terminal: Local x +
There was 1 error:

1) Testing_connection::test_conec
TypeError: Test\Connection::connect(): Return value must be of type PDO, int returned

C:\xampp\htdocs\Perez Lara\Test\Connection.php:31
C:\xampp\htdocs\Perez Lara\Test\Testing_connection.php:10

ERRORS!
Tests: 1, Assertions: 0, Errors: 1.
PS C:\xampp\htdocs\Perez Lara>
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



### Error de retorno corregido

```
Testing_connection.php x
1  <?php
2  use Test\Connection;
3  require ('Connection.php');
4
5  class Testing_connection extends \PHPUnit\Framework\TestCase
6  {
7
8  public function test_conec()
9  {
10     $conec=(new Connection()) → connect();
11     $this→assertIsObject($conec, message: "Es de tipo objeto o no");
12 }
13
```

Prueba a la clase Connection, donde se ejecuta una función (connect) para validar el modelo Connection.

```
Testing_connection.php x
1  use Test\Connection;
2  require ('Connection.php');
3
4  class Testing_connection extends \PHPUnit\Framework\TestCase
5  {
6
7  public function test_conec()
8  {
9     $conec=(new Connection()) → connect();
10     $this→assertIsObject($conec, message: "Es de tipo objeto o no");
11 }
12
13
```

```
Terminal: Local x + v
PS C:\xampp\htdocs\Perez Lara> ./vendor/bin/phpunit Test/Testing_connection.php
PHPUnit 9.5.23 #StandWithUkraine

.
1 / 1 (100%)

Time: 00:00.012, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\xampp\htdocs\Perez Lara>
```