



Servicio Nacional de Aprendizaje SENA

**Centro Electricidad, Electrónica y
Telecomunicaciones**



PRUEBAS UNITARIAS Pérez Lara Cía LTDA.

**SENA
2022**



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



CONTENIDO

INTRODUCCIÓN	2
OBJETIVO PRUEBAS DE SOFTWARE	3
OBJETIVOS ESPECÍFICOS:	3
CARACTERÍSTICAS DE UNA BUENA PRUEBA UNITARIA;	3
PROCESO DE INSTALACIÓN COMPOSER;	4
VERIFICACION DE INSTALACION;	6
PROCESO DE INSTALACIÓN DE PHPUnit	8
REQUISITOS;	8
PRUEBAS UNITARIAS	11
Testing Cliente;	11
Prueba con PHPUnit clase cliente.	12



Servicio Nacional de Aprendizaje SENA

**Centro Electricidad, Electrónica y
Telecomunicaciones**



INTRODUCCIÓN

Este documento da a conocer el plan de pruebas del software que busca comprobar el funcionamiento adecuado del sistema teniendo en cuenta los requerimientos funcionales y no funcionales, por otra parte.

Una prueba unitaria será tan buena como su cobertura de código. La cobertura de código marca la cantidad de código de la aplicación que está sometido a una prueba. Por tanto, si la cobertura es baja, significa que gran parte de nuestro código está sin probar.

OBJETIVO PRUEBAS DE SOFTWARE

Desarrollar un manual para aplicar pruebas unitarias durante el ciclo de desarrollo del proyecto.

OBJETIVOS ESPECÍFICOS:

- Especificar la forma de aplicar pruebas unitarias, en todo el ciclo del desarrollo del proyecto para crear software de calidad.
- Explicar las diferentes metodologías para aplicar pruebas unitarias en el ciclo de desarrollo del proyecto buscando evitar fallos y defectos que retrasen el desarrollo del proyecto y por ende aumente su costo.
- Construir un manual que sea de fácil acceso y claro utilizar las pruebas unitarias en el ciclo de desarrollo de un proceso de implementación para el proyecto.



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



CARACTERÍSTICAS DE UNA BUENA PRUEBA UNITARIA;

Las pruebas unitarias se tienen que poder ejecutar sin necesidad de intervención manual. Esta característica posibilita que podamos automatizar su ejecución.

Las pruebas unitarias tienen que poder repetirse tantas veces como uno quiera. Por este motivo, la rapidez de las pruebas tiene un factor clave. Si pasar las pruebas es un proceso lento no se pasarán de forma habitual, por lo que se perderán los beneficios que éstas nos ofrecen.

Las pruebas unitarias tienen que poder ejecutarse independientemente del estado del entorno. Las pruebas tienen que pasar en cualquier ordenador del equipo de desarrollo.

La ejecución de una prueba no puede afectar la ejecución de otra. Después de la ejecución de una prueba el entorno debería quedar igual que estaba antes de realizar la prueba.

Las diferentes relaciones que puedan existir entre módulos deben ser simulada para evitar dependencias entre módulos.



PROCESO DE INSTALACIÓN COMPOSER;

<https://getcomposer.org/download/>

- El instalador requiere que tengas PHP ya instalado ,descarga Composer por ti y configura tu variable de entorno PATH para que puedas simplemente llamar a Composer desde cualquier directorio.

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

Download Composer

Latest: v2.4.1

Windows Installer

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

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01d4ba75c60ee0459970c2200a51f492d557530c71c15d8dba01eae') {
    echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

This installer script will simply check some `php.ini` settings, warn you if they are set incorrectly, and then download the latest `composer.phar` in the current directory. The 4 lines above will, in order:

- ```
php -r "copy('https://getcomposer.org/installer',
'composer-setup.php');"
php -r "if (hash_file('sha384',
'composer-setup.php') ===
'55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01d4ba75c60ee0459970c2200a51f492d557530c71c15d8dba01eae') {
 echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



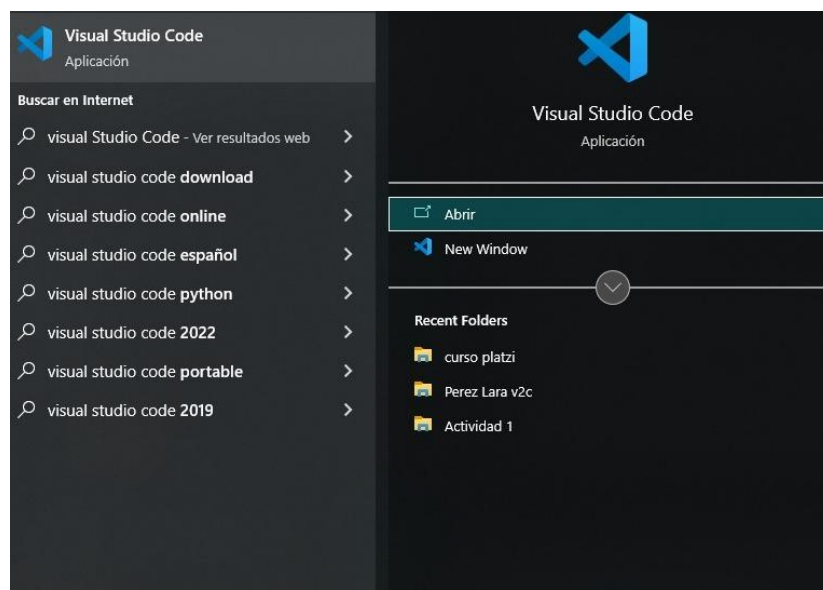
```
PS E:\curso platzi> php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
>> php -r "if (hash_file('sha384', 'composer-setup.php') === '55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01d4ba75c0ee0459970c2200a51f492
d557530c71c15d8dba01eae') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
Installer verified
All settings correct for using Composer
Downloading...

Composer (version 2.4.1) successfully installed to: E:\curso platzi\composer.phar
Use it: php composer.phar

PS E:\curso platzi>
```

## VERIFICACION DE INSTALACION;

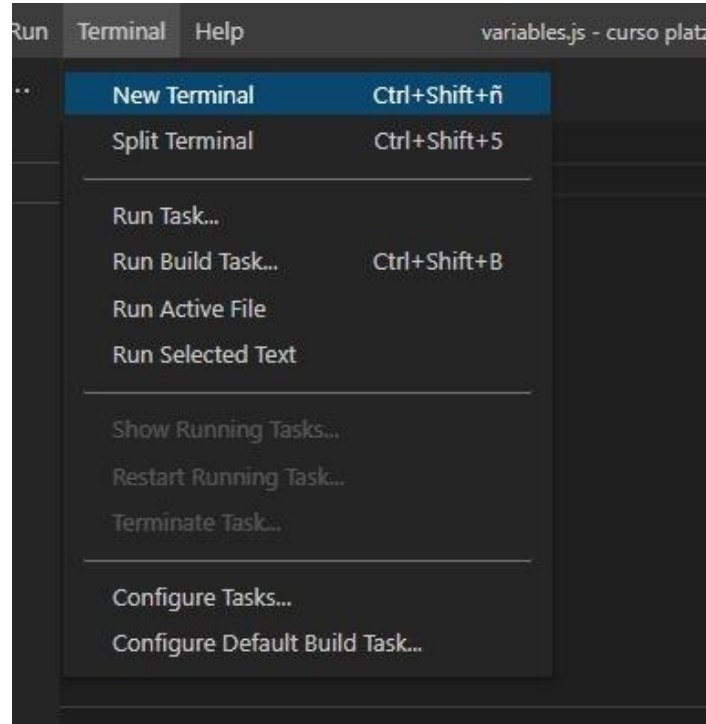
- Abrir en Visual Studio Code
- Nueva Terminal
- Editar
- Verificar C:\composer.phar
- php composer-setup.php





Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS E:\curso platzi>
```

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS E:\curso platzi> php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
>> php -r "if (hash_file('sha384', 'composer-setup.php') === '55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01d4ba75c0ee04
59970c2200a51f492d557530c71c15d8dba01eae') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('com
poser-setup.php'); } echo PHP_EOL;"
>> php composer-setup.php
>> php -r "unlink('composer-setup.php');"
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y  
Telecomunicaciones



```
PS E:\curso platzi> php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
>> php -r "if (hash_file('sha384', 'composer-setup.php') === '55ce33d7678c5a611085589f1f3dd8f8b3c52d662cd01d4ba75c0ee0459970c2200a51f492
d557530c71c15d8dba01eae') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;
"
>> php composer-setup.php
>> php -r "unlink('composer-setup.php');"
Installer verified
All settings correct for using Composer
Downloading...
|
```

## PROCESO DE INSTALACIÓN DE PHPUnit

### REQUISITOS;

- PHPUnit latest requiere PHP 7.3. Se recomienda encarecidamente usar la última versión de PHP.
- PHPUnit requiere las extensiones DOM y JSON que normalmente están habilitadas por defecto.
- Además, PHPUnit requiere las extensiones PREC, y SPL. Estas extensiones estándares se habilitan por defecto y no se pueden desactivar.

- **composer require --dev phpunit/phpunit -w**





## Centro Electricidad, Electrónica y Telecomunicaciones

```
PS E:\curso platzi> composer require --dev phpunit/phpunit -w
Info from https://repo.packagist.org: #StandWithUkraine
Using version ^9.5 for phpunit/phpunit
./composer.json has been created
Running composer update phpunit/phpunit --with-dependencies
Loading composer repositories with package information
Updating dependencies
Lock file operations: 28 installs, 0 updates, 0 removals
- Locking doctrine/instantiator (1.4.1)
- Locking myclabs/deep-copy (1.11.0)
- Locking nikic/php-parser (v4.14.0)
- Locking phar-io/manifest (2.0.3)
- Locking phar-io/version (3.2.1)
- Locking phpunit/php-code-coverage (9.2.17)
- Locking phpunit/php-file-iterator (3.0.6)
- Locking phpunit/php-invoker (3.1.1)
- Locking phpunit/php-text-template (2.0.4)
- Locking phpunit/php-timer (5.0.3)
- Locking phpunit/phpunit (9.5.24)
- Locking sebastian/cli-parser (1.0.1)
- Locking sebastian/code-unit (1.0.8)
- Locking sebastian/code-unit-reverse-lookup (2.0.3)
- Locking sebastian/comparator (4.0.6)
- Locking sebastian/complexity (2.0.2)
- Locking sebastian/diff (4.0.4)
- Locking sebastian/environment (5.1.4)
- Locking sebastian/exporter (4.0.4)
- Locking sebastian/global-state (5.0.5)
- Locking sebastian/lines-of-code (1.0.3)
- Locking sebastian/object-enumerator (4.0.4)
- Locking sebastian/object-reflector (2.0.4)
- Locking sebastian/recursion-context (4.0.4)
- Locking sebastian/resource-operations (3.0.3)
```

```
└─ js-Basico
 ├── JS valores.js
 ├── JS variables.js
 └─ vendor
 ├── bin
 ├── composer
 ├── doctrine
 ├── myclabs
 ├── nikic
 ├── phar-io
 ├── phpunit
 ├── sebastian
 ├── theseer
 ├── autoload.php
 ├── composer.json
 ├── composer.lock
 └── composer.phar
```



```
- Installing sebastian/version (3.0.2): Extracting archive
- Installing sebastian/type (3.1.0): Extracting archive
- Installing sebastian/resource-operations (3.0.3): Extracting archive
- Installing sebastian/recursion-context (4.0.4): Extracting archive
- Installing sebastian/object-reflector (2.0.4): Extracting archive
- Installing sebastian/object-enumerator (4.0.4): Extracting archive
- Installing sebastian/global-state (5.0.5): Extracting archive
- Installing sebastian/exporter (4.0.4): Extracting archive
- Installing sebastian/environment (5.1.4): Extracting archive
- Installing sebastian/diff (4.0.4): Extracting archive
- Installing sebastian/comparator (4.0.6): Extracting archive
- Installing sebastian/code-unit (1.0.8): Extracting archive
- Installing sebastian/cli-parser (1.0.1): Extracting archive
- Installing phpunit/php-timer (5.0.3): Extracting archive
- Installing phpunit/php-text-template (2.0.4): Extracting archive
- Installing phpunit/php-invoker (3.1.1): Extracting archive
- Installing phpunit/php-file-iterator (3.0.6): Extracting archive
- Installing theseer/tokenizer (1.2.1): Extracting archive
- Installing nikic/php-parser (v4.14.0): Extracting archive
- Installing sebastian/lines-of-code (1.0.3): Extracting archive
- Installing sebastian/complexity (2.0.2): Extracting archive
- Installing sebastian/code-unit-reverse-lookup (2.0.3): Extracting archive
- Installing phpunit/php-code-coverage (9.2.17): Extracting archive
- Installing phar-io/version (3.2.1): Extracting archive
- Installing phar-io/manifest (2.0.3): Extracting archive
- Installing myclabs/deep-copy (1.11.0): Extracting archive
- Installing doctrine/instantiator (1.4.1): Extracting archive
- Installing phpunit/phpunit (9.5.24): Extracting archive
7 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating autoload files
25 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
PS E:\curso platzi> |
```



## PRUEBAS UNITARIAS

- **Testing Cliente;**

Se realizan tres Test a tres funciones diferentes un CREATE, UPDATE y DELETE.

```
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 $cel1='311000';
15 $cel2='315000';
16 $direc1='c112';
17 $direc2='c113';
18 $tdocum=2;
19 $ciudad=5;
20 $test=((new Cliente($documento,$correo,$direc1,$direc2,$cel1,$cel2,$nombre1,$nombre2,$apellido1,$apellido2,$tdocum,
21 $fecha,$ciudad))->create());
22 $this->assertEquals(1,$test);
23 }
24
25 public function test_update(){
26 $documento='5151';
27 $documento2='5151';
28 $nombre1='Juan';
29 $nombre2='David';
30 $apellido1='Lopez';
31 $apellido2='Leal';
32 $fecha='2022-06-02';
33 $correo='lopez@gmail.com';
34 $cel1='311000';
35 $cel2='315000';
36 $direc1='c112';
37 $direc2='c113';
38 $tdocum=2;
39 $ciudad=5;
40 $test=((new Cliente($documento,$correo,$direc1,$direc2,$cel1,$cel2,$nombre1,$nombre2,$apellido1,$apellido2,
41 $tdocum,$fecha,$ciudad))->update($documento2));
42 $this->assertEquals(1,$test);
43 }
44
45 public function test_delete()
46 {
47 $documento = '5151';
48
49 $test = ((new Cliente($documento))->delete());
50 $this->assertEquals(1,$test);
51 }
52 }
```



- **Prueba con PHPUnit clase cliente.**

La prueba realizada con PHPUnit a la clase cliente, se realiza con tres funciones diferentes con 3 aserciones correctas.

```
Testing_cliente.php
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 $apellidol1='lopez';
11 $apellidol2='leal';
12 $fecha='2022-06-02';
13 $correo='lopez@gmail.com';
14 $cel1='311000';
15 $cel2='315000';
16 $direc1='c112';
17 $direc2='c113';
18 $tdocum=2;
19 $ciudad=5;
20 $test=((new Cliente($documento,$correo,$direc1,$direc2,$cel1,$cel2,$nombre1,$nombre2,$apellidol1,$apellidol2,$tdocum,
21 $fecha,$ciudad))->create());
22 $this->assertEquals(expected: 1,$test);
23 }
24
25 public function test_update(){
26 $documento='5151';
27 $documento2='5151';
28 $nombre1='JUAN';
29 $nombre2='David';
30 $apellidol1='lopez';
31 $apellidol2='leal';
32 $fecha='2022-06-02';
33 $correo='lopez@gmail.com';
34 $cel1='311000';
35 $cel2='315000';
36 $direc1='c112';
37 $direc2='c113';
38 $tdocum=2;
39 $ciudad=5;
40 $test=((new Cliente($documento,$correo,$direc1,$direc2,$cel1,$cel2,$nombre1,$nombre2,$apellidol1,$apellidol2,
41 $tdocum,$fecha,$ciudad))->update($documento2));
42 $this->assertEquals(expected: 1,$test);
43 }
44
45 public function test_delete()
46 {
47 $documento = '5151';
48
49 $test = ((new Cliente($documento))>delete());
50 $this->assertEquals(expected: 1, $test);
51 }
52 }
```

```
Terminal: Local x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

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**





**Servicio Nacional de Aprendizaje SENA**

**Centro Electricidad, Electrónica y  
Telecomunicaciones**





**Servicio Nacional de Aprendizaje SENA**

**Centro Electricidad, Electrónica y  
Telecomunicaciones**

