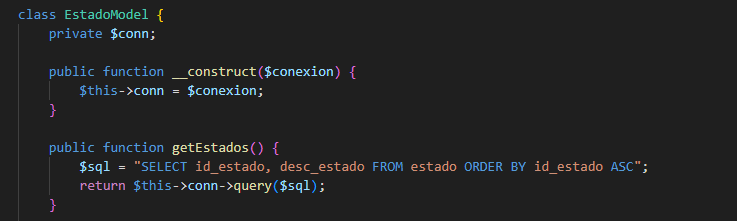
**Manual Pruebas Unitarias**

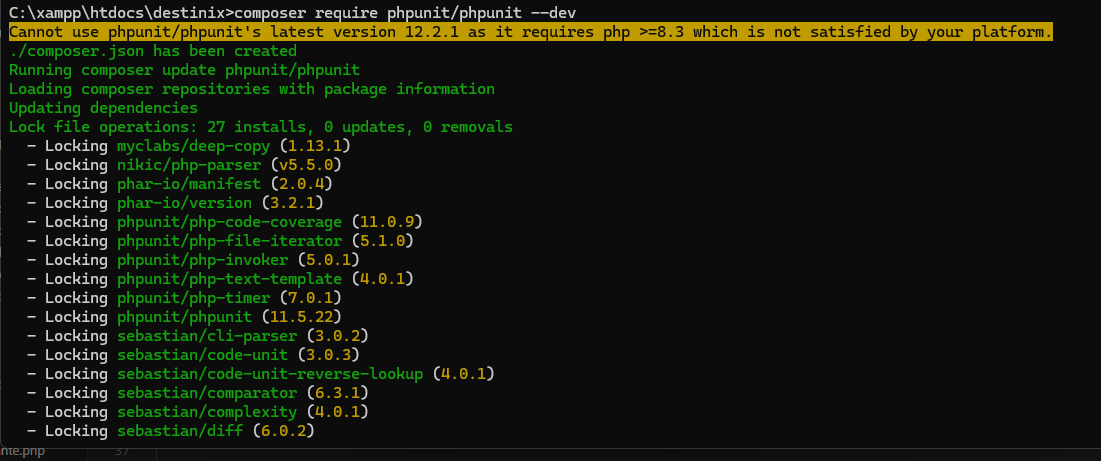
El presente manual tiene como objetivo describir el proceso de implementación y ejecución de pruebas unitarias en el sistema. Las pruebas unitarias permiten verificar que cada función o componente del código cumpla correctamente su propósito de forma aislada. Al seguir este manual, se asegura una mayor confiabilidad del software, facilitando la detección temprana de errores y mejorando el mantenimiento del proyecto.  
  
  
**Instrucciones**

Para realizar una prueba unitaria a un código, primero se debe saber que lenguaje se utiliza, ya que para cada lenguaje se usa una herramienta distinta pero el proceso es muy similar, para este manual utilizaremos **PHP** para realizar las prueba.   
  
Principalmente debemos seleccionar una parte del código al cual deseemos realizarle una prueba unitaria.   
  
para este ejemplo tomaremos una función la cual me trae los estados de la tabla **“Estado”** de mi Base de datos.



una vez seleccionado procedemos a instalar las herramientas que necesitamos, como utilizaremos **PHP** será necesario utilizar **PHPUNIT.**  
  
Ejecutamos la consola y nos ubicamos en la raíz del proyecto donde tenemos los

archivos .**PHP**

(**NOTA:** para moverse entre carpetas en la consola se utiliza “**cd + carpeta a la cual se desea entrar**” y “**cd ..** ” para salir de una carpeta ).  
  
  
Una vez ubicados en la raíz del proyecto , ingresamos este código   
(**composer require phpunit/phpunit --dev** )  
  
de esta forma instalaremos el **PHPUNIT.**  
  
  
Una vez instalado , se crearan 2 archivos y una carpeta



**Composer.json** podremos ver la versión   
Texto

El contenido generado por IA puede ser incorrecto.

y en el **composer.lock** veremos dependencias Texto

El contenido generado por IA puede ser incorrecto.

Ahora la carpeta que se crea es **“Vendor”**  
  
esta contiene una gran cantidad de carpetas que contiene las librerías que necesita el proyecto.  
  
procedemos a tomar la fracción de código que decidimos que le realizaremos la prueba unitaria para enviarlo a un archivo externo , en mi caso lo ubicare dentro de una carpeta llamada **“SRC”**   
  
  
Lo que se realiza es abrir el PHP de esta forma **(<?php)** y abrimos una clase , llamada de la misma forma que el archivo , realizamos la conexión con una base de datos que fue creada para realizar el testeo sin realizar cambios significativos en la base de datos fuente.   
  
y después de eso pegamos nuestra fracción de código al que le realizaremos la prueba unitaria.  
  
  
  
  
  
  
una vez hecho.   
  
crearemos otro archivo, que ubicaremos en una carpeta en la raíz del proyecto que llamamos **“Test”** y dentro de ella creamos un archivo donde realizaremos el testeo , el nombre que utilice fue **“EstadoModelGetTest.php”.**  
  
Una vez con el archivo creado lo que haremos es direccionar el archivo con el anteriormente creado **(EstadoModel)** para luego crear una clase con un nombre distinto , agregando la palabra test podemos mantener un orden en las clases que utilizamos y el testeo que se realiza   
  
y luego realizamos una función que nos va a traer los datos utilizando **“mock ”**  
  
Texto

El contenido generado por IA puede ser incorrecto.  
  
  
ahora para finalizar , ejecutaremos el testeo desde la consola ubicándonos en la raíz del proyecto. Y utilizaremos el código **(vendor\bin\phpunit tests\MiArchivoTest.php)**En mi caso utilizare **“vendor\bin\phpunit tests\EstadoModelGetTest.php”**

**Texto

El contenido generado por IA puede ser incorrecto.**Como podemos ver el testeo fue realizado con éxito.