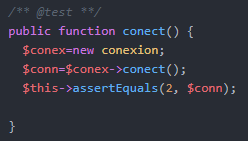
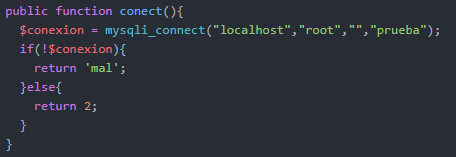
Pruebas con PHPUnit

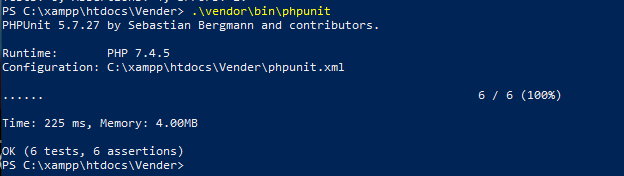
# **Prueba de conexión a BD**

En esta prueba deseamos saber si la conexión con la base de datos es satisfactoria ya que sin ella no podremos hacer los CRUD´S





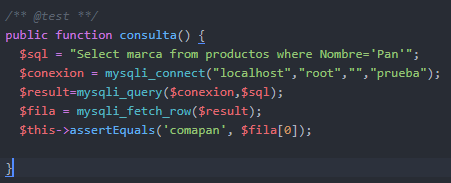
|  |  |  |  |
| --- | --- | --- | --- |
| **Prueba** | **Condición** | **Parámetros** | **Resultado** |
| **#1** | **conexion:**  esta clase es la que nos permitirá hacer uso de la función **conect()** | **conect():**  función que nos devuelve un entero 2 si la conexión fue establecida correctamente | 2 |



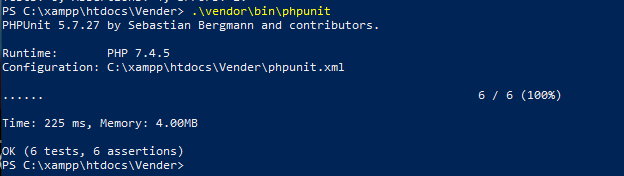
Todas las pruebas han sido realizadas sin ningún fallo, ni error

# **Prueba Consulta de datos**

Verificamos que podemos hacer una consulta de los datos que se encuentran en nuestra base de datos



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prueba** | **Condición** | **Parámetros** | **Objeto** | **Resultado** |
| **#2** | **consulta():**  Método que hace una consulta a la BD, retorna el campo | **$sql:** consulta sql  **$result:** variable en la que guardamos el resultado de nuestra consulta sql a la BD  **$fila:** guarda el resultado de nuestra consulta por filas |  | comapan |

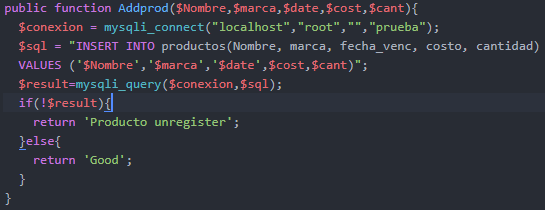


Todas las pruebas han sido realizadas sin ningún fallo, ni error

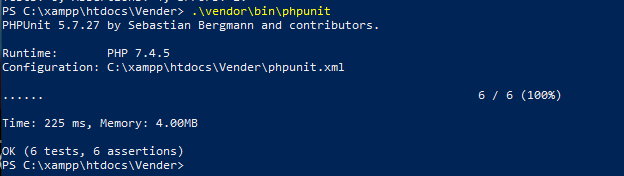
# **Prueba Inserción de datos**

Verificaremos que la inserción de los datos en la base de datos se realiza de manera satisfactoria





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prueba** | **Condición** | **Parámetros** | **Objeto** | **Resultado** |
| **#3** | **insercion():**  Método que retorna un String, si la inserción ha sido realizada exitosamente | **$sql:** consulta sql  **$result:** variable en la que guardamos el resultado de nuestra consulta sql a la BD  **$conn:** guarda el resultado del método **Addprod()**  **Addprod():** método al que se le pasan los datos del producto | **Addprod(**'Arepa','Doña lola','2022-01-29',3800,2**)**; | Good |

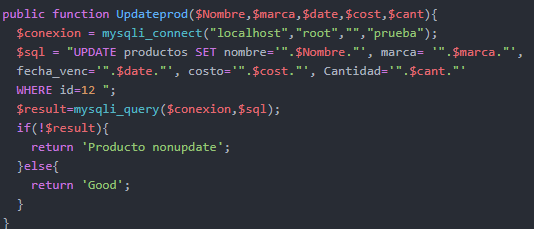


Todas las pruebas han sido realizadas sin ningún fallo, ni error

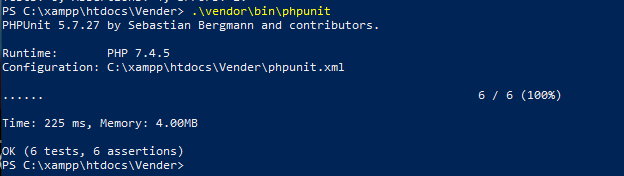
# **Prueba Actualización de datos**

Verificaremos que la actualización de los datos en la base de datos se realiza de manera satisfactoria





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prueba** | **Condición** | **Parámetros** | **Objeto** | **Resultado** |
| **#4** | **actualizacion():**  Método que retorna un String, si la actualización ha sido realizada exitosamente | **$sql:** consulta sql  **$result:** variable en la que guardamos el resultado de nuestra consulta sql a la BD  **$conn:** guarda el resultado del método **Updateprod()**  **Updateprod():** método al que se le pasan los datos del producto | **Updateprod(**'Arepa','Doña Petra,'2023-01-29',3800,2**)**; | Good |

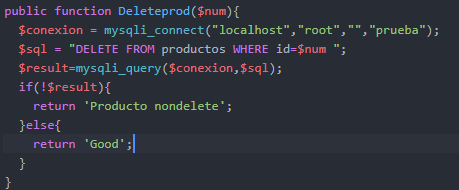


Todas las pruebas han sido realizadas sin ningún fallo, ni error

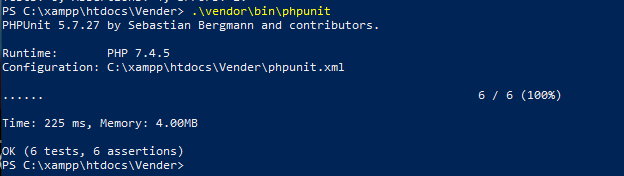
# **Prueba Eliminación de datos**

Verificaremos que la eliminación de los datos en la base de datos se realiza de manera satisfactoria





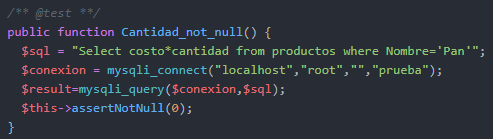
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prueba** | **Condición** | **Parámetros** | **Objeto** | **Resultado** |
| **#5** | **Eliminar():**  Método que retorna un String, si la eliminación ha sido realizada exitosamente | **$sql:** consulta sql  **$result:** variable en la que guardamos el resultado de nuestra consulta sql a la BD  **$conn:** guarda el resultado del método **Deleteprod()**  **Deleteprod():** se le pasa el id del producto a eliminar | **Deleteprod(**14**)**; | Good |



Todas las pruebas han sido realizadas sin ningún fallo, ni error

# **Prueba de cantidad no nula**

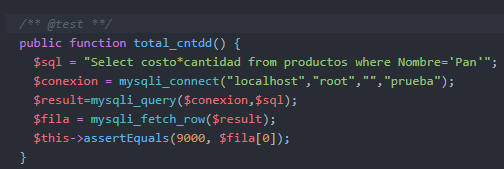
Verificamos que el resultado del precio de un producto no sea nulo



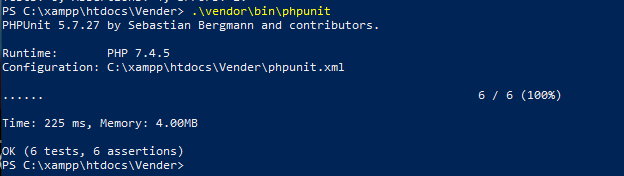
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prueba** | **Condición** | **Parámetros** | **Objeto** | **Resultado** |
| **#6** | **Cantidad\_not\_null():**  Método que retorna un booleano para el resultado de una consulta | **$sql:** consulta sql  **$result:** guarda el resultado de la consulta sql  **assertNotNull():** si el proceso anterior es nulo o no |  | True |

# **Prueba de cantidad**

Verificamos que el resultado de la suma de un producto sea igual al esperado



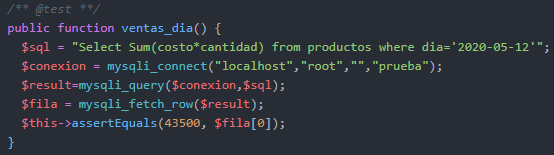
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prueba** | **Condición** | **Parámetros** | **Objeto** | **Resultado** |
| **#7** | **Total\_cntdd():**  Método que retorna un campo especifico de una consulta sql, en este caso todo lo vendido del producto pan | **$sql:** consulta sql  **$result:** guarda el resultado de la consulta sql  **$fila:** obtenemos el resultado separado por filas  **$fila[0]:** especificamos que fila queremos |  | 9000 |



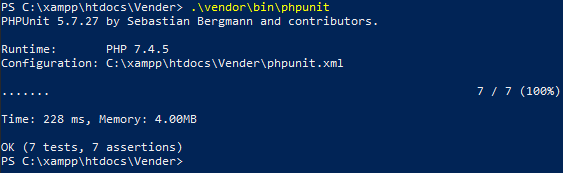
Todas las pruebas sin ningún fallo, ni error

# **Prueba ventas diarias**

Verificamos el resultado de las ventas realizadas de un día en especifico



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prueba** | **Condición** | **Parámetros** | **Objeto** | **Resultado** |
| **#8** | **Ventas\_dia():**  Método que retorna un campo especifico de una consulta sql, en este caso la suma total de las ventas del dia | **$sql:** consulta sql  **$result:** guarda el resultado de la consulta sql  **$fila:** obtenemos el resultado separado por filas  **$fila[0]:** especificamos que fila queremos |  | 43500 |



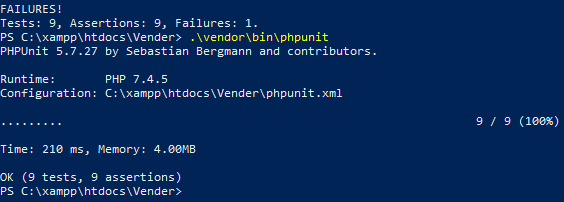
Todas las pruebas siguen sin ningún fallo o error

# **Prueba Index existe**

Verificamos que el archivo este creado en nuestra carpeta del proyecto



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prueba** | **Condición** | **Parámetros** | **Objeto** | **Resultado** |
| **#9** | **Exist\_Index():**  Método que retorna un booleano | **$assertFileExists:** donde vamos a pasar la ruta en la cual debe estar nuestro archivo |  | True |



# **Prueba flujo**

Verificamos el flujo de nuestra aplicación que pase correctamente a cada una de nuestras pestañas y que si hay un botón de regresar o volver funcione correctamente y nos devuelva a nuestra página anterior