



GONZALEZ ROJAS LUIS ALBERTO  
OLASCOAGA RAMOS ÁLVARO SAÚL  
RADILLO RAMOS MIRNA PAOLA  
GRUPO: 8IT12

- Mongodb win32-x86\_64 2008plus ssl 4.0.0
- Xampp version v3.2.2.
- Sublime Text 3 x64
- Driver: php\_mongodb.dll x86 version: 7.2
- Composer

1. Descargar el driver de la siguiente página tomando en cuenta la versión y la arquitectura de PHP.

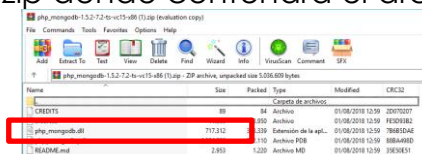
1. Para ver la versión de PHP y la arquitectura debes dirigirte a "Localhost" y dar clic en "PHPInfo".

A screenshot of the Windows Task Manager Performance tab. The 'System' section is highlighted in blue. The 'System' row shows 'Windows NT DESKTOP-EOV4H5S 10.0 build 17134 (Windows 10) x64'. The 'Build Date' row shows 'May 23 2018 20:26:19'. The 'Compiler' row shows 'MSVC15 Visual C++ 2017'. The 'Architecture' row shows 'x64'. The 'PHP Version 7.2.6' is highlighted with a red box in the top left corner. The 'php' logo is in the top right corner.

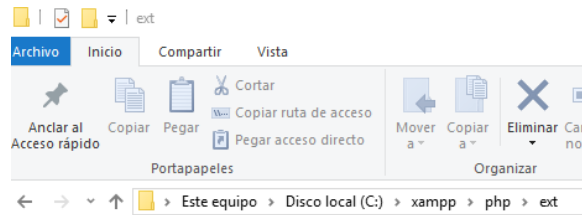
2. Si es Windows debes entrar a la página “[pecl.php.net/package/mongodb/1.52/windows](http://pecl.php.net/package/mongodb/1.52/windows)” y buscar el driver según la versión de PHP y su arquitectura.

DLL List	
PHP 7.2	7.2 Non Thread Safe (NTS) x64 7.2 Thread Safe (TS) x64 7.2 Non Thread Safe (NTS) x86 7.2 Thread Safe (TS) x86

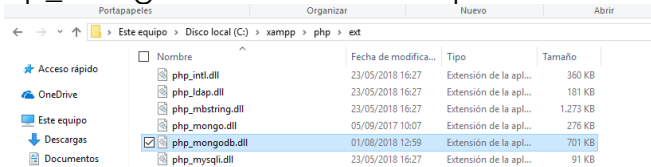
3. Se descargará un archivo .zip donde contendrá el archivo “php\_mongodb.dll”.



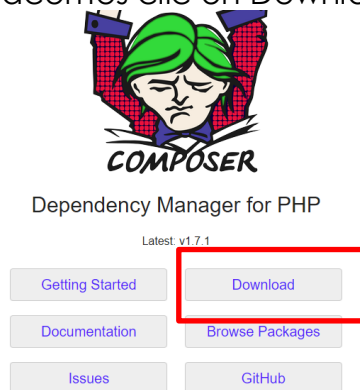
4. Ahora te diriges a esta dirección en tu Explorador de archivos de Windows:  
C:\xampp\php\ext.



5. Extraes el archivo "php\_mongodb.dll" en esa carpeta.



6. Para instalar las librerías de Composer nos dirigimos a la página  
"https://getcomposer.org/" y hacemos clic en Download.



7. Después clic en "Composer-Setup.exe" y se descargará el archivo para después ejecutarlo.

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

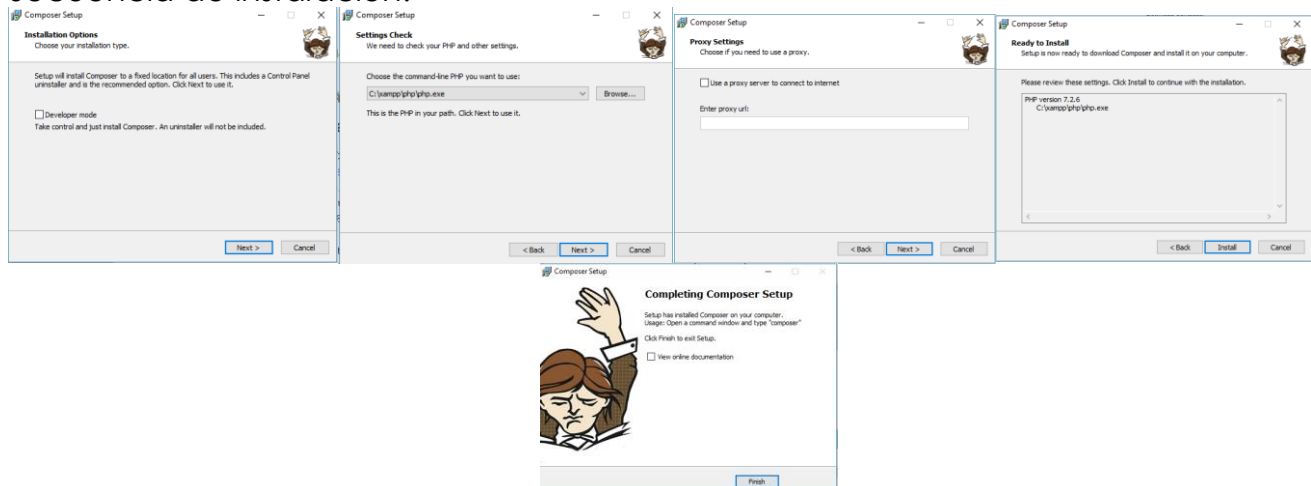
Download Composer Latest: v1.7.1

#### Windows Installer

The installer will download composer for you and set up your PATH environment variable so you can simply cd directory

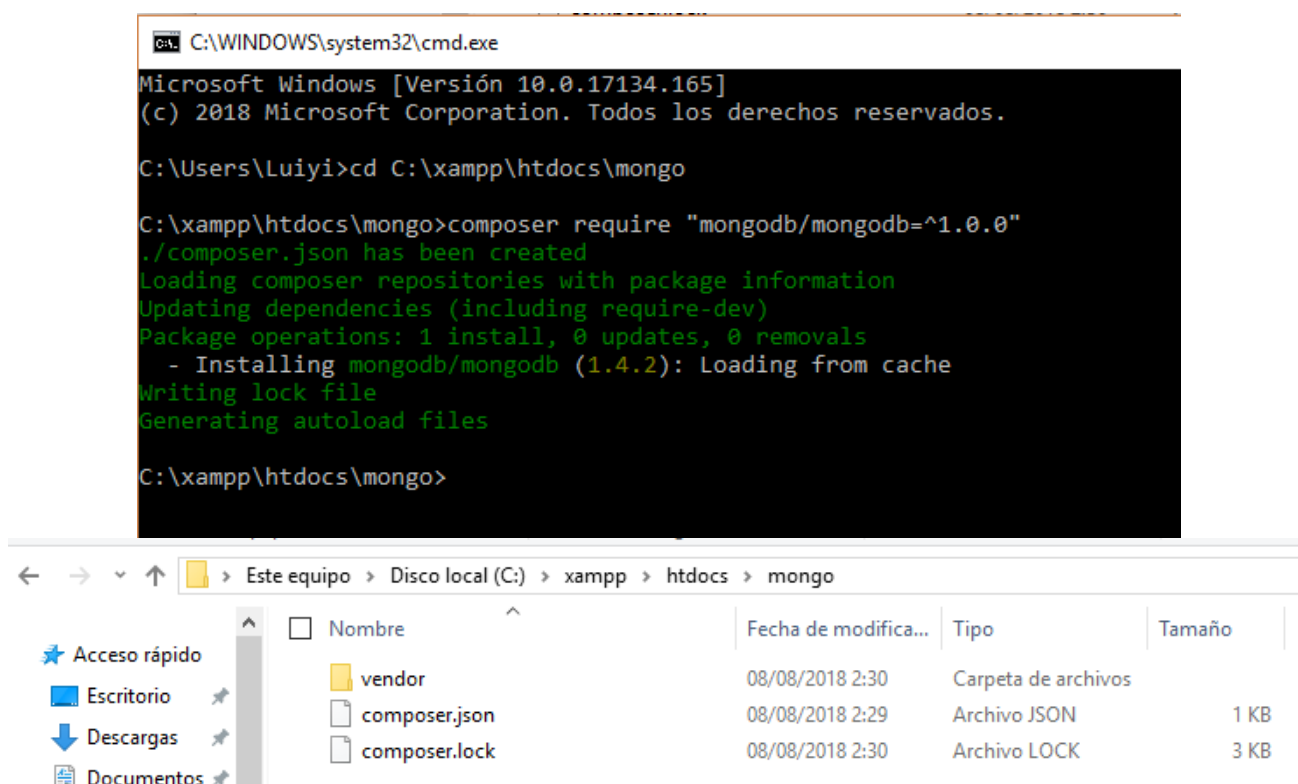
Download and run **Composer-Setup.exe** - it will install the latest composer version whenever it is executed.

#### Secuencia de instalación:

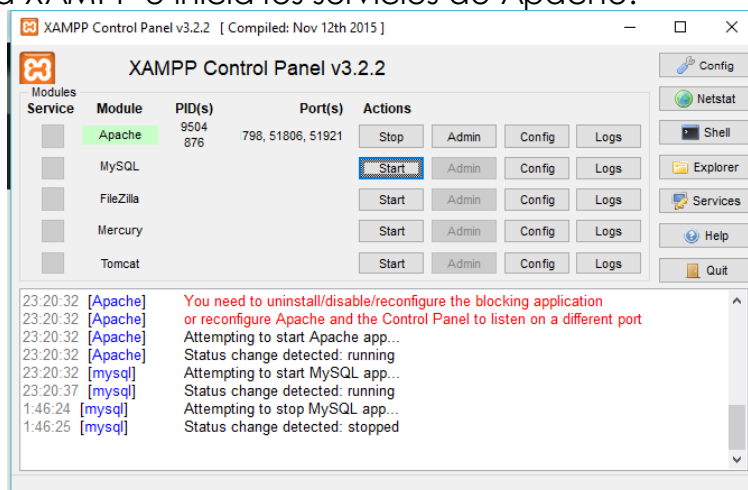


8. Abrimos cmd oprimiendo Windows+r y colocamos los siguientes codigos secuencialmente:

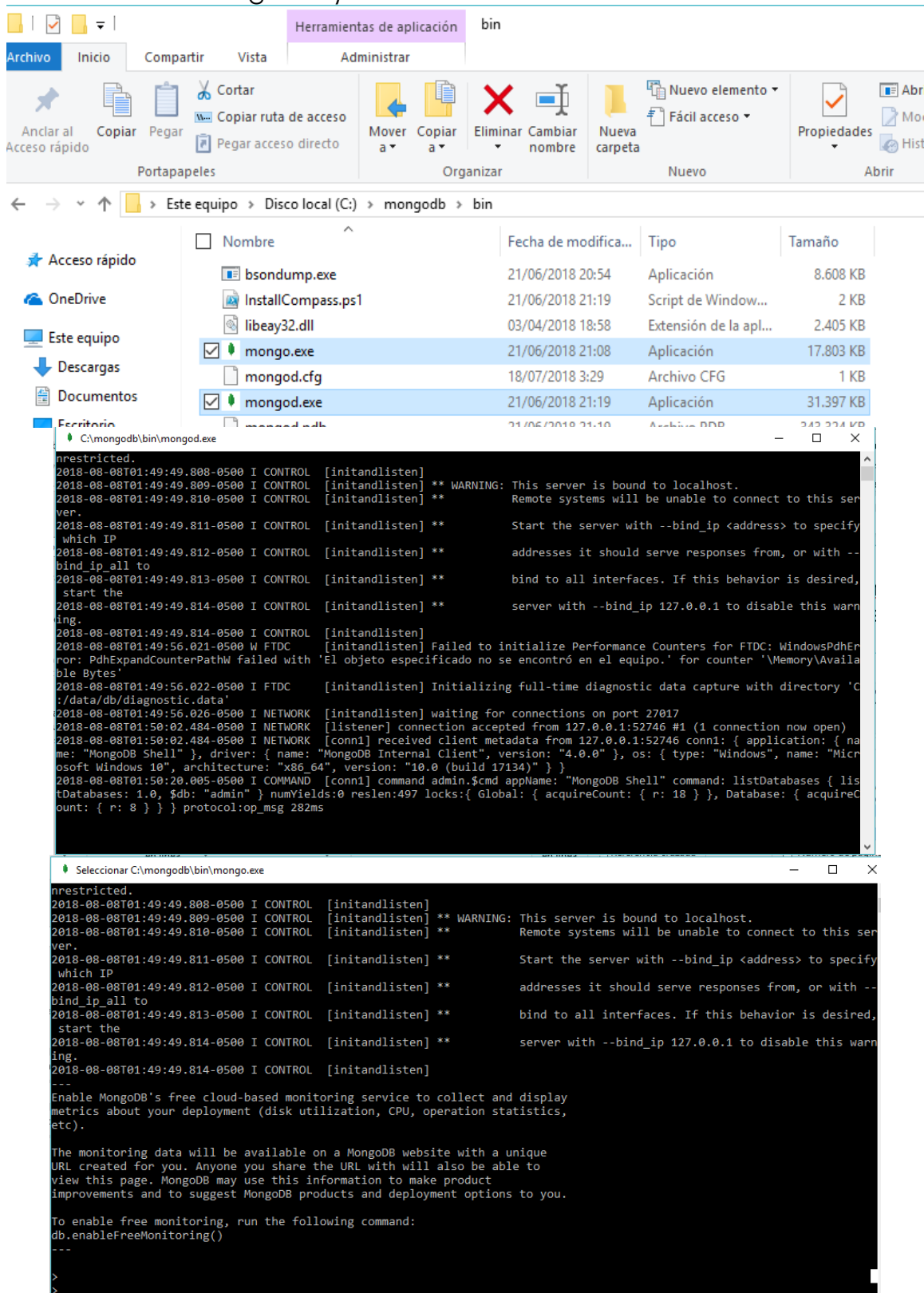
1. `cd C:\xampp\htdocs\mongo`: para especificar la carpeta que usaremos.
2. `composer require "mongodb/mongodb=^1.0.0"`: para extraer los archivos de composer en la carpeta seleccionada.
3. Por último, nos dirigimos a la carpeta que seleccionamos y vemos que ya están los archivos de composer.



9. Ahora se ejecuta XAMPP e inicia los servicios de Apache.



10. Iniciamos MongoDB dirigiéndonos a la dirección "C:\mongodb\bin" y ejecutamos con doble clic los archivos "mongod.exe" y "mongo.exe" en ese orden (en ocasiones se ejecuta dos veces mongo.exe).



## CODIFICACIÓN:

### 1. Conexión a MongoDB desde el archivo "conexion.php".

El siguiente código muestra:

1. Se importan las funciones y métodos del archivo autoload.php en la carpeta vendor.
2. Se instancia una variable llamada \$mongo para crear la conexión.
3. En la variable \$db se guarda y selecciona la base de datos llamada "tienda", si no existe se crea.
4. En caso de error se recupera el mensaje del error y se imprime

```
conexion.php
1  <?php
2  try{
3      require 'vendor/autoload.php';
4          $mongo = new \MongoDB\Client;
5          $db = $mongo->selectDatabase('tienda');
6      }
7      catch(MongoConnectionException $e){
8
9          die($e->getMessage());
10     };
11  ?>
```

### 2. Inserción de datos desde el archivo "menú\_productos.php" donde se almacena el diseño y "insertar\_productos2.php" donde se encuentra el código php para poder insertar los datos en la colección.

Formulario insertar productos

```
menu-productos.php
77  <form action="insertar_productos2.php" method="post" name="form1" >
78      <div class="col-md-6">
79          <div class="form-group">
80              <label for="nombre">Nombre:</label><br>
81              <input class="form-control" type="text" name="nombre" id="nombre" placeholder="Nombre del producto *"
82                  data-validation-required-message="Por favor introduce el nombre del producto." required>
83              <p class="help-block text-danger"></p>
84          </div>
85          <div class="form-group">
86              <label for="descripcion">Descripcion:</label><br>
87              <input class="form-control" type="text" name="descripcion" id="descripcion" placeholder="Descripcion del producto *"
88                  data-validation-required-message="Por favor introduce La descripción del producto." required>
89              <p class="help-block text-danger"></p>
90          </div>
91          <div class="form-group">
92              <label for="precio">Precio:</label><br>
93              <input class="form-control" type="number" name="precio" id="precio" placeholder="Precio del producto *"
94                  data-validation-required-message="Por favor introduce el precio del producto." required>
95              <p class="help-block text-danger"></p>
96          </div>
97          <div>
98              <br><br>
99              <td>
100                 <button class="btn btn-primary btn-xl text-uppercase js-scroll-trigger" type="submit" name="Submit"> Registrar </button>
101             </td>
102         </div>
103     </div>
104 </form>
```

## Código para insertar productos

El siguiente código muestra:

1. Importación de la conexión a la base de datos.
2. Si se oprime el botón Submit: se guardan en un array los datos de nombre, descripción y precio insertados en el anterior formulario.
3. Se inserta el array por medio de la sentencia "insertOne".

```
insertar_productos2.php
63 <?php
64 include_once("conexion.php");
65 if(isset($_POST['Submit'])) {
66     $producto = array (
67         'nombre' => $_POST['nombre'],
68         'descripcion' => $_POST['descripcion'],
69         'precio' => $_POST['precio']
70     );
71     $db->productos->insertOne($producto);
72 }
73 ?>
74
```

3. **Consulta de datos** desde el archivo "menú\_productos.php" dentro de una colección generada por el código almacenado en el mismo archivo.

El siguiente código muestra:

1. Diseño de la estructura gráfica.
2. Importación de la conexión a la base de datos.
3. Selección y almacenamiento de la colección productos en la variable \$col\_productos.
4. Consulta y almacenamiento de la sentencia find().
5. Creación de los encabezados de la tabla.
6. Consulta de cada uno de los documentos con sus respectivos atributos y con botones a un lado para modificar o eliminar ese documento.

```
menu_productos.php
111 <!-- EMPIEZA CONSULTAR PRODUCTOS -->
112 <section id="portfolio">
113     <div class="container">
114
115         <div class="col-lg-12 text-center">
116             <h2 class="section-heading text-uppercase">Consultar Productos</h2>
117             <h3 class="section-subheading text-muted">Productos dados de alta hasta el momento</h3>
118         </div>
119
120         <div class="text-center">
121             <div class="col-lg-12">
122                 <CENTER>
123
124
125                 <?php
126                 include("conexion.php");
127                 $col_productos = $db->productos;
128                 $result = $col_productos->find();
129
130                 echo "<table><th>Nombre      </th><th>Descripción      </th><th>Precio      </th><th></th><th></th></th>";
131                 foreach ($result as $entry) {
132                     echo "<tr><td>'. $entry['nombre']. '      </td>';
133                     echo "<td>'. $entry['descripcion']. '      </td>';
134                     echo "<td>'. $entry['precio']. '      </td>';
135                     echo "<td><a href=modificar_productos.php?id='.$entry['_id']. '><button id='sendMessageButton' class='btn btn-primary btn-xl
136                     text-uppercase' type='button'>Modificar</button></a></td>";
137                     echo "<td><a href=eliminar_producto.php?id='.$entry['_id']. '><button id='sendMessageButton' class='btn btn-primary btn-xl
138                     text-uppercase' type='button'>Eliminar</button></a></td>";
139                 }
140                 echo "</table>";
141             }
142         }
143     }
144 </div>
145 </section>
146 ?>
```



#### 4. Actualización de documentos.

Mediante un botón para modificar en la tabla de consultas se ejecuta el siguiente código.

En el siguiente código se consulta los datos del documento buscados por el id del producto:

```
55 <?php
56
57 include("conexion.php");
58 $id=$_GET["id"];
59 $col_productos = $db->productos;
60 $result = $col_productos->findOne(['_id'=> new MongoDB\BSON\ObjectID($id)]);
61
62 ?>
```

El diseño del formulario y los datos del documento consultado se muestran en el siguiente código:

```
71 <form id='form' name='form' novalidate='novalidate' action='modificar_productos2.php' method='post'>
72 <div class='col-md-6'>
73 <div class='form-group'>
74 Id del producto: <br><input class='form-control' placeholder='ID del producto *' data-validation-required-message='
75 Por favor introduce la descripción del producto.' type='text' name='id' readonly='readonly' value= "<?php echo $
76 result['_id'] ?>"></input>
77 </div>
78 <div class='form-group'>
79 Nombre:<br><input class='form-control' placeholder='Nombre del producto *' data-validation-required-message='Por
80 favor introduce el nombre del producto.' required type='text' name='nombre'
81 value= "<?php echo $result['nombre'] ?>"></input>
82 </div>
83 <div class='form-group'>
84 Descripción: <br><input class='form-control' placeholder='Descripción del producto *' data-validation-required-message
85 = 'Por favor introduce la descripción del producto.' required type='text' name='descripcion' value= "<?php echo $
86 result['descripcion'] ?>"></input>
87 </div>
88 <div class='form-group'>
89 Precio:<br><input class='form-control' placeholder='Precio del producto *' data-validation-required-message='Por
90 favor introduce el precio del producto.' required type='number' name='precio'
91 value= "<?php echo $result['precio'] ?>"></input>
92 </div>
93 <button class='btn btn-primary btn-xl text-uppercase js-scroll-trigger'> Modificar </button>
94 </div>
95 </form>
```

El siguiente código muestra:

1. El mensaje del documento actualizado.
2. Importación de la conexión a la base de datos
3. Se guardan en un array los datos de nombre, descripción y precio insertados en el anterior formulario.
4. Se consulta y almacena en una variable llamada \$condicion el id obtenido del producto,
5. Se ejecuta la sentencia para actualizar el producto con los parámetros de condición y el array que contiene los datos del producto mediante la sentencia updateOne().

```
53 <!-- Header -->
54
55 <header class="masthead">
56 <div class="container">
57 <div class="intro-text">
58 <div class="intro-lead-in">Registro modificado!</div>
59 <a class="btn btn-primary btn-xl text-uppercase js-scroll-trigger" href="menu-productos.php#services">Volver a
60 modificar un producto</a>
61 </div>
62 </div>
63 </header>
64
65 <?php
66
67 include_once("conexion.php");
68 $collection = $db->productos;
69 $producto = array (
70 'set' => array(
71 'nombre' => $_POST['nombre'],
72 'descripcion' => $_POST['descripcion'],
73 'precio' => $_POST['precio']
74 ));
75 $condicion=array("_id" => new MongoDB\BSON\ObjectID($_POST['id']));
76 $db->productos->updateOne($condicion,$producto);
77
78 ?>
```

## 5. Eliminación de documentos.

Mediante un botón para eliminar en la tabla de consultas se ejecuta el siguiente código y muestra:

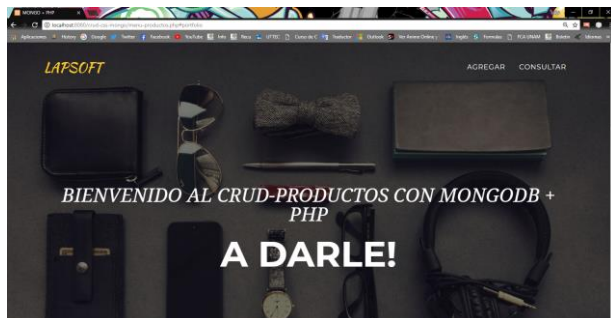
1. *Importación de la conexión a la base de datos*
2. Se recupera el id del documento seleccionado
3. Se selecciona y almacena en una variable la colección productos.
4. Se eliminar el documento dependiendo del id almacenado anteriormente,
5. Se ejecuta un Script que lanza una ventana emergente para avisar que el documento ha sido eliminado.
6. Se actualiza el archivo "menú\_productos.php".

```
eliminar_producto.php
1 <?php
2 include("conexion.php");
3
4 $id=$_GET['id'];
5 $col_producto = $db->productos;
6
7 $result = $col_producto->deleteOne(['_id'=> new MongoDB\BSON\ObjectID($id)]);
8
9 ?>
10 <script type="text/javascript">
11     alert("Producto eliminado correctamente");
12     window.location.href='menu-productos.php#portfolio';
13 </script>
```

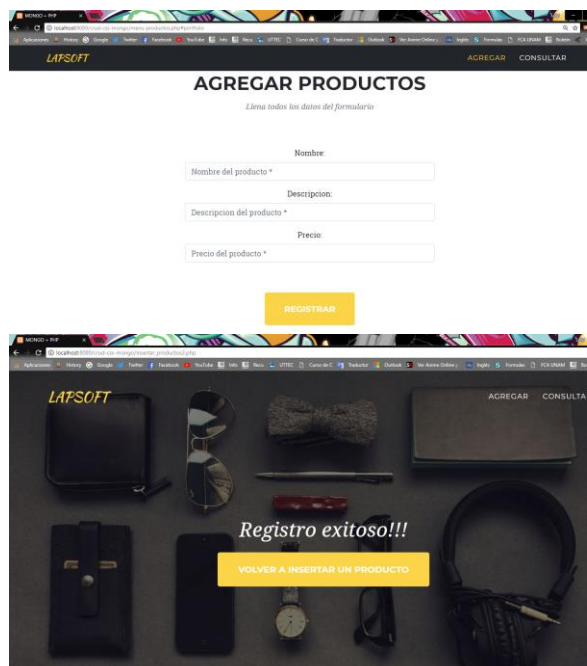


## INTERFACES GRÁFICAS:

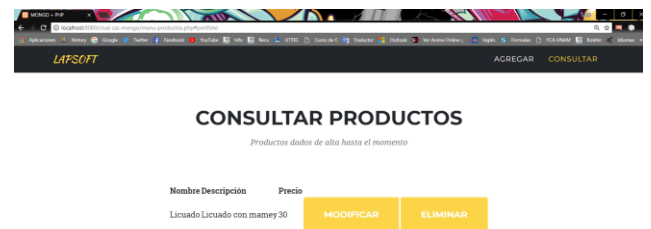
### 1. Interfaz principal



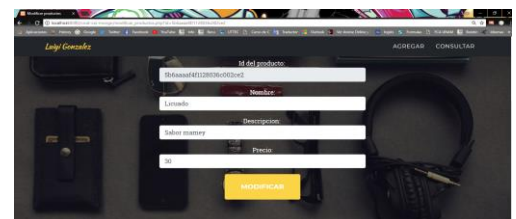
### 2. Agregar productos.



### 3. Consultar productos.



### 4. Modificar productos.



### 5. Eliminar productos.

