



En esta práctica desarrollaremos un CRUD en php utilizando PDO y arquitectura MVC.

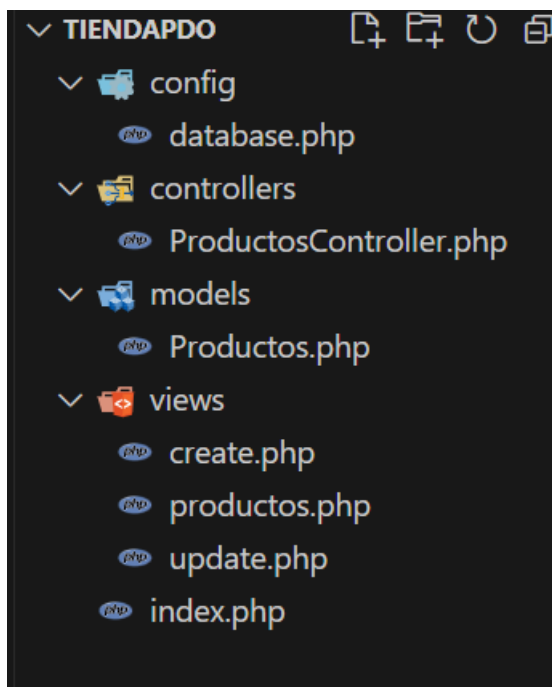
PDO (Php Data Objects) es una extensión de php que proporciona una interfaz uniforme para acceder a diferentes bases de datos.

Ventajas:

- **Portabilidad:** Puedes cambiar de base de datos sin modificar mucho tu código.
- **Seguridad:** Las sentencias preparadas ayudan a prevenir inyecciones SQL.
- **Flexibilidad:** Soporta múltiples bases de datos y ofrece varias opciones de configuración.

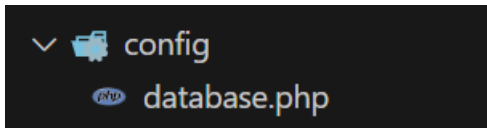
Arquitectura MVC (Modelo-Vista-Controlador) es un patron de diseño de software que se utiliza para separar la lógica de la aplicación en tres componentes interrelacionados pero distintos. Esto facilita la gestión y el mantenimiento del código, así como la colaboración entre desarrolladores.

Ahora iniciaremos la práctica de esta semana. Para ello crearemos la siguiente estructura en nuestro proyecto llamado TiendaPDO.





Como primero paso configuraremos la conexión con el servidor y la base de datos, para ello nos situaremos en el archivo config/database.php

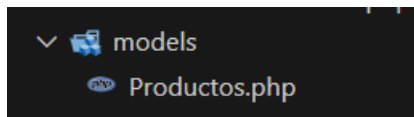


Y codificaremos lo siguiente:

```
config > database.php > Database
1  <?php
2
3  1 reference | 0 implementations
   class Database{
4      1 reference
       private $host = "localhost"; // nombre del servidor
5      1 reference
       private $database = "dbtienda"; // nombre de la base de datos
6      1 reference
       private $user = "root"; // nombre del usuario
7      1 reference
       private $password = ""; // contraseña
8      4 references
       public $conn; // atributo auxiliar para realizar la conexión
9
10     // Metodo para establecer la conexión |
11     1 reference | 0 overrides
       public function getConnection(){
12         $this->conn = null;
13         try {
14             $this->conn = new PDO(
15                 "mysql:host=" . $this->host . ";dbname=" . $this->database .
16                 $this->user, $this->password);
17
18             $this->conn->exec("set names utf8");
19         } catch (PDOException $e) {
20             echo "Connection error " . $e->getMessage();
21             die();
22         }
23         return $this->conn;
24     }
25 }
```



Como siguiente paso crearemos nuestro modelo



Codificaremos el siguiente código:

```
models > Productos.php > Productos > get_product_by_id()
1  <?php
2
3  1 reference | 0 implementations
4  class Productos{
5      6 references
6      private $conn;
7      5 references
8      private $table_name = "productos"; // nombre de la tabla
9
10     // atributos que hacen referencia a los campos de la tabla
11     8 references
12     public $id;
13     9 references
14     public $name;
15     9 references
16     public $description;
17     9 references
18     public $categorie;
19
20     1 reference | 0 overrides
21     public function __construct($db){
22         $this->conn = $db;
23     }
24
25     // Metodo para crear un producto
26     1 reference | 0 overrides
27     public function create(){
28         //Creamos la consulta
29         $query = "INSERT INTO " . $this->table_name .
30         " SET name=:name, description=:description, categorie=:categorie";
31         $result = $this->conn->prepare($query);
32
33         //Limpiamos el codigo
34         $this->name = htmlspecialchars(strip_tags($this->name));
35         $this->description = htmlspecialchars(strip_tags($this->description));
36         $this->categorie = htmlspecialchars(strip_tags($this->categorie));
37
38         $result->bindParam(":name", $this->name);
39         $result->bindParam(":description", $this->description);
40         $result->bindParam(":categorie", $this->categorie);
41
42         if ($result->execute()){
43             return true;
44         }
45         else{
46             return false;
47         }
48     }
49 }
```



```
41 // Metodo para obtener todos los productos
    1 reference | 0 overrides
42 public function get_products(){
43     $query = "SELECT * FROM " . $this->table_name;
44     $result = $this->conn->prepare($query);
45     $result->execute();
46     return $result;
47 }
48
```

```
    1 reference | 0 overrides
49 public function get_product_by_id(){
50     $query = "SELECT * FROM " . $this->table_name .
51     " WHERE id = :id";
52     $result = $this->conn->prepare($query);
53     $result->bindParam("id", $this->id);
54     $result->execute();
55
56     $row = $result->fetch(PDO::FETCH_ASSOC);
57
58     $this->name = $row["name"];
59     $this->description = $row["description"];
60     $this->categorie = $row["categorie"];
61 }
62
```

```
63 // Metodo para actualizar un producto
    1 reference | 0 overrides
64 public function update(){
65     $query = "UPDATE " . $this->table_name .
66     " SET name=:name, description=:description, categorie=:categorie" .
67     " WHERE id=:id";
68
69     $this->name = htmlspecialchars(strip_tags($this->name));
70     $this->description = htmlspecialchars(strip_tags($this->description));
71     $this->categorie = htmlspecialchars(strip_tags($this->categorie));
72     $this->id = htmlspecialchars(strip_tags($this->id));
73
74     $result = $this->conn->prepare($query);
75     $result->bindParam("name", $this->name);
76     $result->bindParam("description", $this->description);
77     $result->bindParam("categorie", $this->categorie);
78     $result->bindParam("id", $this->id);
79
80     if ($result->execute()){
81         return true;
82     }
83     else{
84         return false;
85     }
86 }
```

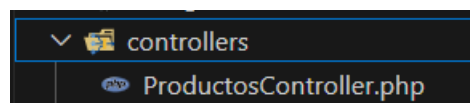


```

88      // Metodo para eliminar un producto
      1 reference | 0 overrides
89      public function delete(){
90          $query = "DELETE FROM ". $this->table_name .
91              " WHERE id=:id";
92          $result = $this->conn->prepare($query);
93          $result->bindParam("id", $this->id);
94          if ($result->execute()){
95              return true;
96          }
97          else{
98              return false;
99          }
100     }
101
102 }

```

Ahora nos ubicaremos en el controlador



```

1  <?php
2  require_once(dirname(__FILE__) . '/../config/database.php');
3  require_once(dirname(__FILE__) . '/../models/Productos.php');
4
5  1 reference | 0 implementations
6  class ProductosController{
7      2 references
8      private $db;
9      15 references
10     private $producto;
11
12     1 reference | 0 overrides
13     public function __construct(){
14         // capturamos la conexion a la bd
15         $database = new Database();
16         $this->db = $database->getConnection();
17         // Instanciamos el modelo productos
18         $this->producto = new Productos($this->db);
19     }
20
21     // Metodo para mostrar la lista de productos
22     3 references | 0 overrides
23     public function index(){
24         $result = $this->producto->get_products();
25         $productos = $result->fetchAll(PDO::FETCH_ASSOC);
26
27         // Llamamos la vista que muestra la lista de productos
28         include(dirname(__FILE__) . '/../views/productos.php');
29     }
30 }

```

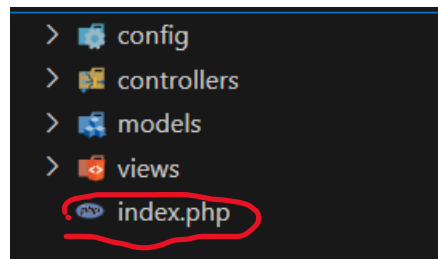


```
26 // Metodo para crear un nuevo usuario
27 1 reference | 0 overrides
28 public function create(){
29     if($_POST){
30         // Asignamos los datos del formulario a las propiedades del usuario
31         $this->producto->name = $_POST['name'];
32         $this->producto->description = $_POST['description'];
33         $this->producto->categoria = $_POST['categoria'];
34
35         header("Location: index.php");
36         return $this->producto->create();
37     }
38
39     // Incluimos la vista del formulario de creacion
40     include(dirname(__FILE__) . '/../views/create.php');
41 }
42
43 // Metodo para editar producto
44 1 reference | 0 overrides
45 public function edit($id){
46     $this->producto->id = $id;
47     $this->producto->get_product_by_id();
48
49     if ($_POST){
50         $this->producto->id = $id;
51         $this->producto->name = $_POST['name'];
52         $this->producto->description = $_POST['description'];
53         $this->producto->categoria = $_POST['categoria'];
54
55         header("Location: index.php");
56         return $this->producto->update();
57     }
58
59     include(dirname(__FILE__) . '/../views/update.php');
```

```
60 // Metodo para eliminar producto
61 1 reference | 0 overrides
62 public function delete($id){
63     $this->producto->id = $id;
64     header("Location: index.php");
65     return $this->producto->delete();
66 }
```



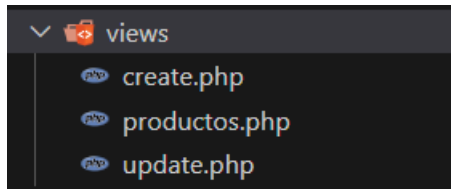
Una vez creado el modelo y el controlador configuraremos el archivo index.php



```
index.php
1  <?php
2
3  require_once '../controllers/ProductosController.php';
4
5  // Capturamos la opcion de la URL
6  $action = isset($_GET['action']) ? $_GET['action'] : '';
7  $id = isset($_GET['id']) ? $_GET['id'] : '';
8
9  // Instanciamos el controlador
10 $controller = new ProductosController();
11
12 // Switch para manejar las direcciones basadas en el valor $action
13 switch ($action) {
14     case 'create':
15         $controller->create();
16         break;
17     case 'edit':
18         if ($id){
19             $controller->edit($id);
20         }
21         else {
22             $controller->index();
23         }
24         break;
25     case 'delete':
26         if ($id){
27             $controller->delete($id);
28         }
29         else {
30             $controller->index();
31         }
32         break;
33     default:
34         $controller->index();
35         break;
36 }
```



Por último, nos centraremos en las vistas



Primero en productos.php

```
<body>
  <div class="container">
    <div class="card m-auto mt-5 p-4">
      <h2>Lista de productos</h2>
      <div class="row">
        <div class="col-md-3">
          <a href="index.php?action=create" type="button" class="btn btn-success">Agregar</a>
        </div>
      </div>
      <div class="row">
        <table class="table">
          <thead class="thead-dark">
            <th scope="col">Nombre</th>
            <th scope="col">Descripcion</th>
            <th scope="col">Categoria</th>
            <th scope="col">Opciones</th>
          </thead>
          <tbody>
            <?php foreach($productos as $producto) : ?>
              <tr>
                <td><?= htmlspecialchars($producto['name']); ?></td>
                <td><?= htmlspecialchars($producto['description']); ?></td>
                <td><?= htmlspecialchars($producto['categoria']); ?></td>
                <td>
                  <a href="index.php?action=edit&id=<?= $producto['id'] ?>">Editar</a>
                  <a href="index.php?action=delete&id=<?= $producto['id'] ?>">Eliminar</a>
                </td>
              </tr>
            <?php endforeach; ?>
          </tbody>
        </table>
      </div>
    </div>
  </div>
</body>
```




Create.php

```
<body>
  <div class="container">
    <form action="" method="post" name="frmCreate">
      <div class="row mt-5">
        <div class="col-md-3">
          <label for="">Nombre:</label>
          <input type="text" name="name" class="form-control">
        </div>
        <div class="col-md-3">
          <label for="">Categoria:</label>
          <input type="text" name="categorie" class="form-control">
        </div>
      </div>
      <div class="row">
        <div class="col-md-6">
          <label for="">Descripcion:</label>
          <textarea name="description" id="" class="form-control"></textarea>
        </div>
      </div>
      <div class="row mt-3">
        <div class="col-md-3">
          <button type="submit" class="btn btn-success">Guardar</button>
          <a href="index.php" class="btn btn-danger">Cancelar</a>
        </div>
      </div>
    </form>
  </div>
</body>
```

Update.php

```
<body>
  <div class="container">
    <form action="" method="post">
      <div class="row mt-5">
        <div class="col-md-3">
          <label for="">Nombre:</label>
          <input type="text" name="name" value="{<? htmlspecialchars($this->producto->name) ?}"
            class="form-control">
        </div>
        <div class="col-md-3">
          <label for="">Categoria:</label>
          <input type="text" name="categorie" value="{<? htmlspecialchars($this->producto->categorie) ?}"
            class="form-control">
        </div>
      </div>
      <div class="row">
        <div class="col-md-6">
          <label for="">Descripcion:</label>
          <textarea name="description" id="" class="form-control">{<? htmlspecialchars
            ($this->producto->description) ?}</textarea>
        </div>
      </div>
      <div class="row mt-3">
        <div class="col-md-3">
          <button type="submit" class="btn btn-success">Guardar</button>
          <a href="index.php" class="btn btn-danger">Cancelar</a>
        </div>
      </div>
    </form>
  </div>
</body>
```



Reto seminal:

Completa la función de actualizar registro.