

DOCUMENTACIÓN

1º- Creación de la base de datos

```
Setting environment for using XAMPP for Windows.
Juan José@DESKTOP-91UP6BN c:\xampp
# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.17-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database crud2;
Query OK, 1 row affected (0.003 sec)
```

Ahora debemos de darle permisos a un usuario para poder trabajar con ella

```
MariaDB [(none)]> grant all on crud2.* to urepaso@'localhost';
Query OK, 0 rows affected (0.004 sec)
```

Creamos la tabla e introducimos los datos

```
MariaDB [crud2]> create table articulos(
  -> id int PRIMARY KEY AUTO_INCREMENT,
  -> nombre VARCHAR(60) unique not null,
  -> pvp decimal(6,2) not null,
  -> stock INT UNSIGNED default 0
  -> );
Query OK, 0 rows affected (0.018 sec)

MariaDB [crud2]> select * from articulos;
Empty set (0.006 sec)

MariaDB [crud2]> insert into articulos(nombre, pvp, stock) values("Monitor 19", 123.45, 12);
Query OK, 1 row affected (0.002 sec)

MariaDB [crud2]> insert into articulos(nombre, pvp, stock) values("Raton USB", 23.95, 13);
Query OK, 1 row affected (0.002 sec)

MariaDB [crud2]> insert into articulos(nombre, pvp, stock) values("USB 512GB", 123.45, 14);
Query OK, 1 row affected (0.002 sec)

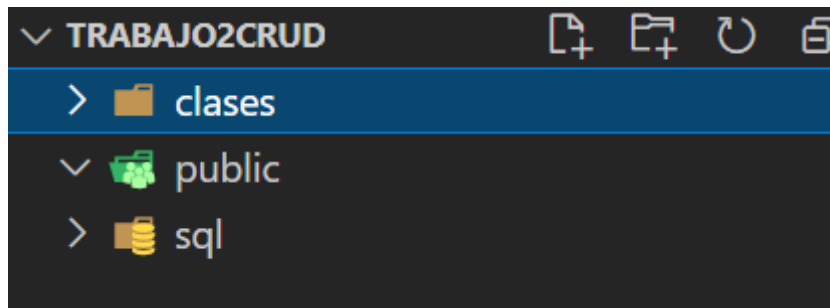
MariaDB [crud2]> insert into articulos(nombre, pvp, stock) values("USB 8GB", 12.45, 14);
Query OK, 1 row affected (0.002 sec)

MariaDB [crud2]> insert into articulos(nombre, pvp, stock) values("SET LAPICEROS", 13.15, 32);
Query OK, 1 row affected (0.002 sec)
```

Comprobamos que los datos se han introducido correctamente

```
MariaDB [crud2]> select * from articulos;
+-----+-----+-----+-----+
| id | nombre | pvp | stock |
+-----+-----+-----+-----+
| 1 | Monitor 19 | 123.45 | 12 |
| 2 | Raton USB | 23.95 | 13 |
| 3 | USB 512GB | 123.45 | 14 |
| 4 | USB 8GB | 12.45 | 14 |
| 5 | SET LAPICEROS | 13.15 | 32 |
| 6 | Stick Wifi | 23.00 | 62 |
| 7 | Funda 17 | 13.40 | 92 |
| 8 | Monitor 22 | 193.65 | 2 |
| 9 | Smart TV 52 | 453.85 | 92 |
| 10 | SSD 512GB | 123.45 | 2 |
+-----+-----+-----+-----+
10 rows in set (0.000 sec)
```

2º.- Creación del proyecto con las carpetas necesarias como son sql, public y clases



Inicializamos el composer init

```
Juan José@DESKTOP-91UP6BN C:\xampp\htdocs\php\repasomayo\trabajo2crud
# composer init

Welcome to the Composer config generator

This command will guide you through creating your composer.json config.

Package name (<vendor>/<name>) [juan jos/trabajo2crud]: juanjose/trabajo2crud
Description []: Segundo trabajo PDO
Author [, n to skip]: n
Minimum Stability []:
Package Type (e.g. library, project, metapackage, composer-plugin) []: project
License []: GNU

Define your dependencies.

Would you like to define your dependencies (require) interactively [yes]?
Search for a package:
Would you like to define your dev dependencies (require-dev) interactively [yes]?
Search for a package:

{
    "name": "juanjose/trabajo2crud",
    "description": "Segundo trabajo PDO",
    "type": "project",
    "license": "GNU",
    "require": {}
}

Do you confirm generation [yes]?
```

Así debe de quedar el composer.json

```
composer.json X
composer.json > {} autoload
1  {
2      "name": "juanjose/trabajo2crud",
3      "description": "Segundo trabajo PDO",
4      "type": "project",
5      "license": "GNU",
6      "require": {},
7      "config": {
8          "optimize-autoloader": true
9      },
10     "autoload": {
11         "psr-4": {
12             "Clases\\": "clases"
13         }
14     }
15 }
16
```

Después de hacer el composer update se nos crea un composer.lock

```
Juan José@DESKTOP-91UP6BN C:\xampp\htdocs\php\repasomayo\trabajo2crud
# composer update
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating optimized autoload files
```

```
composer.json  composer.lock X
composer.lock > ...
1  {
2      "_readme": [
3          "This file locks the dependencies of your project to a single state, so that
4          reproducing this project exactly matches what you have here.",
5          "Read more about it at https://getcomposer.org/doc/01-repositories.md#lock
6          This file is @generated automatically"
7      ],
8      "content-hash": "85dcb91fc48df7ca142b65a888aade0",
9      "packages": [],
10     "packages-dev": [],
11     "aliases": [],
12     "minimum-stability": "stable",
13     "stability-flags": [],
14     "prefer-stable": false,
15     "prefer-lowest": false,
16     "platform": [],
17     "platform-dev": [],
18     "plugin-api-version": "2.0.0"
19 }
```

Creamos el .config para enlazar el proyecto con la base de datos anteriormente creada

```
.config
1  ;parametros de conexion
2  [opciones]
3  host=localhost
4  bbdd=crud2
5  usuario=urepaso
6  pass=secret0
```

3.- Proyecto

Creamos el archivo Conexion.php dentro de la carpeta clases

```
clases > Conexion.php > Conexion
1  <?php
2  namespace Clases;
3  use PDO;
4  use PDOException;
5
6  class Conexion{
7      protected static $conexion;
8
9      public function __construct()
10     {
11         if(self::$conexion==null){
12             self::crearConexion();
13         }
14     }
15     public static function crearConexion(){
16         $opciones=parse_ini_file('../.config');
17         $mibase=$opciones["bbdd"];
18         $miUser=$opciones["usuario"];
19         $miPass=$opciones["pass"];
20         $miHost=$opciones["host"];
21         $dns="mysql:host=$miHost;dbname=$mibase;charset=utf8mb4";
22         try {
23             self::$conexion=new PDO($dns,$miUser,$miPass);
24             self::$conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
25         } catch (PDOException $ex) {
26             die("Error al conectar a la BBDD,mensaje".$ex->getMessage());
27         }
28     }
29 }
```

Ahora crearemos dentro de la misma carpeta el archivo Articulos.php donde meteremos las funciones con las sentencias sql necesarias para el proyecto.

```

clases > Articulos.php > PHP Intelephense > Articulos > leerDatos
1  <?php
2
3  namespace Clases;
4  require "../clases/Conexion.php";
5  use Clases\Conexion;
6  use PDO;
7  use PDOException;
8
9  class Articulos extends Conexion
10 {
11     private $id;
12     private $nombre;
13     private $pvp;
14     private $stock;
15
16     public function __construct()
17     {
18         parent::__construct();
19     }
20
21     public function create()
22     {
23         $i = "insert into articulos(nombre,pvp,stock) values(:n,:p,:s)";
24         $stmt = parent::$conexion->prepare($i);
25         try {
26             $stmt->execute([
27                 ':n' => $this->nombre,
28                 ':p' => $this->pvp,
29                 ':s' => $this->stock
30             ]);
31         } catch (PDOException $ex) {
32             die("Error al crear un articulo:" . $ex->getMessage());
33         }
34     }
35
36     public function mostrarDatos(){
37         $con = "select * from articulos";
38         $stmt = parent::$conexion->prepare($con);
39         try {
40             $stmt->execute();
41         } catch (PDOException $ex) {
42             die("Error al mostrar todos los articulos" . $ex->getMessage());
43         }
44         return $stmt;
45     }
46 }

```

```

public function leerDatos()
{
    $c = "select * from articulos where id=:i";
    $stmt = parent::$conexion->prepare($c);
    try {
        $stmt->execute([
            ':i' => $this->id
        ]);
    } catch (PDOException $ex) {
        die("Error al leer un tag:" . $ex->getMessage());
    }
    $obj = $stmt->fetch(PDO::FETCH_OBJ);
    return $obj;
}

public function existeArticulo($nombre)
{
    $c = "select * from articulos where nombre=:n";
    $stmt = parent::$conexion->prepare($c);
    try {
        $stmt->execute([
            ':n' => $nombre
        ]);
    } catch (PDOException $ex) {
        die("Error al comprobar existencia de articulos:" . $ex->getMessage());
    }
    $fila = $stmt->fetch(PDO::FETCH_OBJ);
    return ($fila == null) ? false : true;
}

public function update()
{
    $u = "update articulos set nombre=:n, pvp=:p, stock=:s where id=:i";
    $stmt = parent::$conexion->prepare($u);
    try {
        $stmt->execute([
            ':i' => $this->id,
            ':n' => $this->nombre,
            ':p' => $this->pvp,
            ':s' => $this->stock
        ]);
    } catch (PDOException $ex) {
        die("Error al editar un articulo:" . $ex->getMessage());
    }
}

```

Mas los getters y setters

Dentro de public creamos una carpeta llamada resources donde crearemos un archivo llamado mensaje.php que usaremos para mostrar los mensajes en las demás páginas

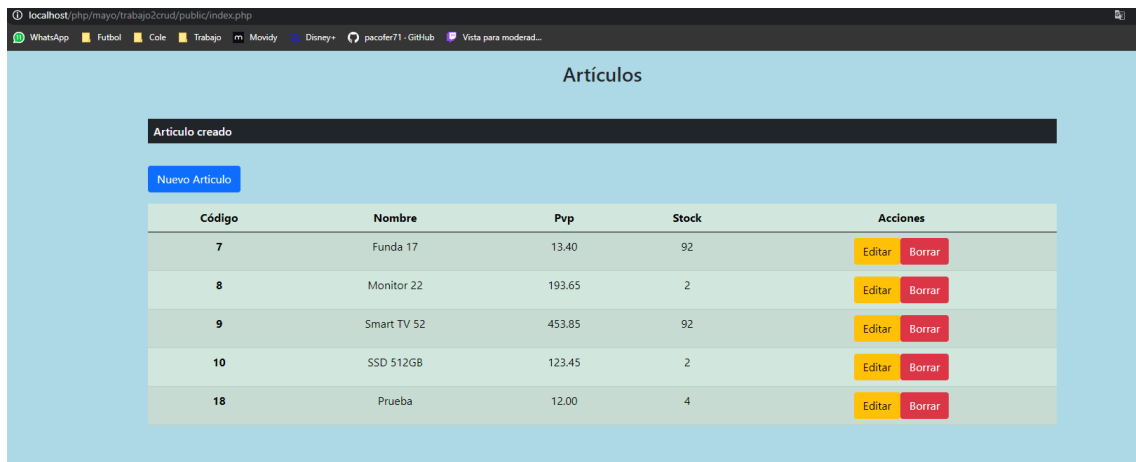
```
public > resources > mensaje.php
1  <?php
2  if(isset($_SESSION['mensaje'])) {
3      echo "<h6 class='bg-dark text-light my-3 p-2'>{$_SESSION['mensaje']}</h6>";
4      unset($_SESSION['mensaje']);
5  }
```

Ahora nos creamos dentro de public un archivo llamado index.php el cual usaremos para mostrar todos los datos de la bbdd

```
public > index.php > ...
1  <?php
2  session_start();
3
4  use Clases\Articulos;
5
6  require '../vendor/autoload.php';
7
8  $esteArticulo = new Articulos();
9  $datos = $esteArticulo->mostrarDatos();
10 $esteArticulo = null;
11 ?>
12 |
13 <!DOCTYPE html>
14 <html lang="en">
15
16 <head>
17     <meta charset="UTF-8">
18     <meta http-equiv="X-UA-Compatible" content="IE=edge">
19     <meta name="viewport" content="width=device-width, initial-scale=1.0">
20     <!-- CSS only -->
21     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-e02MYsds33i+sc0/b76fSiCZc+SNDVn2yr8+0RDqrQ10h+e"
22     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" integrity="sha512-iBBXm8fW90+nuL3CK1bmPCLa00T92x01B1sZ+ymDW2CvqsmGcc"
23     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JE9xM8G8R+phE31jM4H6MPP0WintQrMb457Z0daulmUtxwG2v15DKltS3q"
24     <title>Index</title>
25 </head>
26
27 <body style="background-color: lightblue">
28     <h3 class="text-center mt-3">Articulos</h3>
29     <div class="container mt-5">
30         <?php
31             require 'resources/mensaje.php';
32         ?>
33         <a href="crearArticulos.php" class="btn btn-primary my-3">Nuevo Articulo</a>
34         <table class="table table-success table-striped text-center">
35             <thead>
36                 <tr>
37                     <th scope="col">Código</th>
38                     <th scope="col">Nombre</th>
39                     <th scope="col">Pvp</th>
40                     <th scope="col">Stock</th>
41                     <th scope="col">Acciones</th>
42                 </tr>
43             </thead>
44             <tbody>
45                 <?php
46                     while ($fila = $datos->fetch(PDO::FETCH_OBJ)) {
47                         echo "<tr>";
48                         echo "<th scope='row'>{$fila->id}</th>";
49                         echo "<td>{$fila->nombre}</td>";
50                         echo "<td>{$fila->pvp}</td>";
51                         echo "<td>{$fila->stock}</td>";
52                         echo "<td>";
53                         echo "<form name='as' method='POST' class='form-inline' action='borrarArticulos.php'>\n";
54                         echo "<a href='editarArticulos.php?id={$fila->id}' class='btn btn-warning'>Editar</a>";
55                         echo "<input type='hidden' name='codigo' value='{$fila->id}'>";
56                         echo "<button type='submit' class='ml-2 btn btn-danger'>Borrar</button>\n";
57                         echo "</form>";
58                         echo "</td>";
59                         echo "</tr>";
60                     }
61                 ?>
62             </tbody>
63         </table>
64     </div>
65
66 </body>
```

```
public > index.php > ...
27 <body style="background-color: lightblue">
28     <h3 class="text-center mt-3">Articulos</h3>
29     <div class="container mt-5">
30         <?php
31             require 'resources/mensaje.php';
32         ?>
33         <a href="crearArticulos.php" class="btn btn-primary my-3">Nuevo Articulo</a>
34         <table class="table table-success table-striped text-center">
35             <thead>
36                 <tr>
37                     <th scope="col">Código</th>
38                     <th scope="col">Nombre</th>
39                     <th scope="col">Pvp</th>
40                     <th scope="col">Stock</th>
41                     <th scope="col">Acciones</th>
42                 </tr>
43             </thead>
44             <tbody>
45                 <?php
46                     while ($fila = $datos->fetch(PDO::FETCH_OBJ)) {
47                         echo "<tr>";
48                         echo "<th scope='row'>{$fila->id}</th>";
49                         echo "<td>{$fila->nombre}</td>";
50                         echo "<td>{$fila->pvp}</td>";
51                         echo "<td>{$fila->stock}</td>";
52                         echo "<td>";
53                         echo "<form name='as' method='POST' class='form-inline' action='borrarArticulos.php'>\n";
54                         echo "<a href='editarArticulos.php?id={$fila->id}' class='btn btn-warning'>Editar</a>";
55                         echo "<input type='hidden' name='codigo' value='{$fila->id}'>";
56                         echo "<button type='submit' class='ml-2 btn btn-danger'>Borrar</button>\n";
57                         echo "</form>";
58                         echo "</td>";
59                         echo "</tr>";
60                     }
61                 ?>
62             </tbody>
63         </table>
64     </div>
65
66 </body>
```

Resultado de la página index.php



Código	Nombre	Pvp	Stock	Acciones	
7	Funda 17	13.40	92	Editar	Borrar
8	Monitor 22	193.65	2	Editar	Borrar
9	Smart TV 52	453.85	92	Editar	Borrar
10	SSD 512GB	123.45	2	Editar	Borrar
18	Prueba	12.00	4	Editar	Borrar

Después creamos una pagina llamada crearArticulos.php que usaremos para crear artículos nuevos

```
public > crearArticulos.php > ...
1  <?php
2  session_start();
3  require '../vendor/autoload.php';
4
5  use Clases\Articulos;
6
7
8  if (isset($_POST['crear'])) {
9
10     $catn = trim($_POST['nombre']);
11     $catp = trim($_POST['pvp']);
12     $cats = trim($_POST['stock']);
13
14     $esteArticulo = new Articulos();
15     if (!$esteArticulo->existeArticulo(ucwords($catn))) {
16         $esteArticulo->setNombre(ucwords($catn));
17         $esteArticulo->setPvp(ucwords($catp));
18         $esteArticulo->setStock(ucwords($cats));
19         $esteArticulo->create();
20         $esteArticulo = null;
21         $_SESSION['mensaje'] = "Artículo creado";
22         header("Location:index.php");
23     } else {
24         $_SESSION['mensaje'] = "El artículo ya existe";
25         $esteArticulo = null;
26         header("Location:{$_SERVER['PHP_SELF']}");
27         die();
28     }
29 } else {
30     //pintamos el form
31
32 }
33
34 <!DOCTYPE html>
35 <html lang="en">
36
37 <head>
38     <meta charset="UTF-8">
39     <meta http-equiv="X-UA-Compatible" content="IE=edge">
40     <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
public > crearArticulos.php > ...
35 <html lang="en">
36
37 <head>
38 <meta charset="UTF-8">
39 <meta http-equiv="X-UA-Compatible" content="IE=edge">
40 <meta name="viewport" content="width=device-width, initial-scale=1.0">
41 <!-- CSS only -->
42 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-6K9QAoyanqS54H+/v7LeQ2gyOC3uYarJCUQ7k8J6JCYT/IU6W2j8Qvv8634060" crossorigin="anonymous">
43 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@5.15.3/css/all.min.css" integrity="sha384-Si2q8+8w1y10Yp08U/ktv853U0N13pJ308pAB4XNz8EGW471W/q1VB469Bb44" crossorigin="anonymous">
44 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JSG7Eh3Rk7Cp6KnOPNDJ58lGDsRzq7m0FK1CE4GCsJEO/Rb17XyZvHH+mW0B5T/8" crossorigin="anonymous"></script>
45 <title>Crear Articulos</title>
46 </head>
47
48 <body style="background-color:bisque">
49 <h3 class="text-center mt-3">Crear Articulos</h3>
50 <div class="container mt-3">
51 <?php
52 require 'resources/mensaje.php';
53 ?>
54 <form name="nt" action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
55 <div class="mt-2">
56 <input type="text" name="nombre" placeholder="Nombre" required />
57 </div>
58 <div class="mt-2">
59 <input type="number" name="pvp" placeholder="Pvp" required />
60 </div>
61 <div class="mt-2">
62 <input type="number" name="stock" placeholder="Stock" required />
63 </div>
64 <div class="mt-2">
65 <input type="submit" name="crear" placeholder="Crear" class="btn btn-success mr-2" />
66 <input type="reset" value="Limpiar" class="btn btn-warning mr-2" />
67 <a href="index.php" class="btn btn-primary">Volver</a>
68 </div>
69 </form>
70 </div>
71 </body>
72
73 </html>
74
75 <?php ?>
```

Resultado en el navegador:



Crear Articulos

Nombre

Pvp

Stock

Enviar Limpiar Volver

El siguiente paso será crear la pagina editarArticulos.php dentro de la carpeta public

```
public > editarArticulos.php > ...
1  <?php
2  session_start();
3
4  if (!isset($_GET['id'])) {
5      header("Location:index.php");
6  }
7
8  require '../vendor/autoload.php';
9
10
11  use Clases\Articulos;
12
13  $id = $_GET['id'];
14  $esteArticulo = new Articulos();
15  $esteArticulo->setId($id);
16  $obj = $esteArticulo->leerDatos();
17
18
19
20  if (isset($_POST['editar'])) {
21      //procesamos el formulario
22      $catn = trim($_POST['nombre']);
23      $catp = trim($_POST['pvp']);
24      $cats = trim($_POST['stock']);
25
26
27      if (strlen($catn) == 0) {
28          $_SESSION['mensaje'] = "Rellene el campo";
29          header("Location:{$_SERVER['PHP_SELF']}?id=$id");
30          die();
31      }
32
33      $esteArticulo->setNombre(ucwords($catn));
34      $esteArticulo->setPvp(ucwords($catp));
35      $esteArticulo->setStock(ucwords($cats));
36      $esteArticulo->update();
37      $esteArticulo = null;
38      $_SESSION['mensaje'] = "Articulo editado";
39      header("Location:index.php");
40  } else {
41      //pintamos el form
42
43  ?>
44
45  <!DOCTYPE html>
46  <html lang="en">
47
48  <head>
49      <meta charset="UTF-8">
50      <meta http-equiv="X-UA-Compatible" content="IE=edge">
51      <meta name="viewport" content="width=device-width, initial-scale=1.0">
52      <!-- CSS only -->
53      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rvY096078Vk7w3l04mp1N67e66gwzDUAZF6z3X09" crossorigin="anonymous">
54      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" integrity="sha384-Sf9P8f1q3vr1T6pyt6bK6br/Wzl/Phrg9qprJ2pu4H4vS7NY6k3s+h41tzahhW12Sf4" crossorigin="anonymous">
55      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JX9p6qk60Bb1rLzfQ14Y1qsIZPPtdEPzE+fOrLT6pJvF1W61K1s+F09Jw8" crossorigin="anonymous">
56      <title>Editar Articulos</title>
57  </head>
58
59  <body style="background-color:bisque">
60      <h3 class="text-center mt-3">Editar Articulo</h3>
61      <div class="container mt-3">
62          <!-- NECESITAMOS PASARLE EL ID PARA QUE SEPA CUAL ES EL ID Y NO NOS VUELVA A CARGAR TAGS.PHP POR EL HEADER AL NO PASARLE EL ID -->
63          <form name="nt" action="{$_SERVER['PHP_SELF']}?id=$id" method="POST">
64              <div class="mt-2">
65                  <input type="text" name="nombre" required value="{$_obj->nombre}" class="form-control" />
66                  <input type="text" name="pvp" required value="{$_obj->pvp}" class="form-control" />
67                  <input type="text" name="stock" required value="{$_obj->stock}" class="form-control" />
68              </div>
69              <div class="mt-2">
70                  <input type="submit" name="editar" value="Editar" class="btn btn-success mr-2" />
71                  <a href="index.php" class="btn btn-primary">Volver</a>
72              </div>
73          </form>
74      </div>
75  </body>
76
77  </html>
```

Resultado en el navegador:

Editar Artículo

Funda 17
13.40
92

EditarVolver

Por último debemos de crear la clase borrarArticulos.php dentro de la carpeta public

```
public > borrarArticulos.php > ...
1  <?php
2
3  require "../vendor/autoload.php";
4
5  use Clases\Articulos;
6
7  if(!isset($_POST['codigo'])){
8      header("Location:index.php");
9      die();
10 }
11
12 $esteArticulo = new Articulos();
13 $esteArticulo->setId($_POST['codigo']);
14 $esteArticulo->borrar();
15 $esteArticulo = null;
16 $_SESSION['mensaje']="Artículo Borrado Correctamente";
17 header("Location:index.php");
```