

# **Tareas Unidad 3**

## **Desarrollo Web en Entorno Servidor**

Autor: Derimán Tejera Fumero

Fecha: 11/10/2024

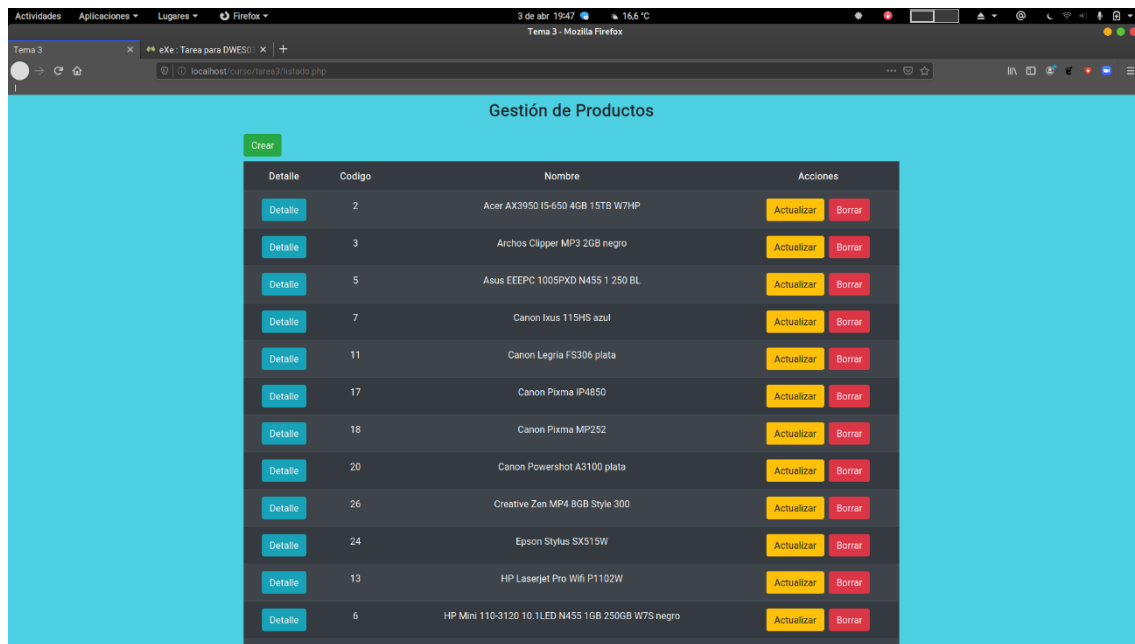
# Índice

Enunciado de la práctica .....	3
Explicación del código .....	6
conexion.php.....	6
listado.php.....	7
detalle.php .....	10
crear.php.....	13
update.php .....	17
borrar.php.....	20
Mostrando la ejecución del CRUD.....	21
listado.php.....	21
crear.php.....	22
Detalle.php.....	23
update.php .....	24
borrar.php.....	25

# Enunciado de la práctica

Partiendo de la base de datos 'proyecto' usada en los ejemplos y ejercicios de la unidad, se trata de programar un CRUD (create, read, update, delete) que permita gestionar los registros de la tabla 'productos'. La aplicación se dividirá en 5 páginas:

- listado.php. Mostrará en una tabla los datos código y nombre y los botones para crear un nuevo registro, actualizar uno existente, borrarlo o ver todos sus detalles. Ver imagen.



- crear.php. Será un formulario para rellenar todos los campos de productos (a excepción del id). Para la familia nos aparecerá un "select" con los nombre de las familias de los productos para elegir uno (lógicamente aunque mostremos los nombres pro formulario enviaremos el código). Ver imagen.

**Crear Producto**

Nombre:

Nombre Corto:

Precio (€):

Descripción:

Familia:   
 Cámaras digitales  
 Cámaras digitales  
 Consolas  
 Equipos multifunción  
 Impresoras  
 Libros electrónicos  
**Memorias flash**  
 Netbooks  
 Ordenadores  
 Ordenadores portátiles  
 Reproductores MP3  
 Routers  
 Sistemas de alimentación ininterrumpida  
 Software  
 Televisores  
 Videocámaras

[Crear](#) [Limpiar](#) [Volver](#)

- detalle.php. Mostrará todos los detalles del producto seleccionado.

**Detalle Producto**

HP Laserjet Pro Wifi P1102W

Código: 13

Nombre: HP Laserjet Pro Wifi P1102W

Nombre Corto: LJPROP1102W

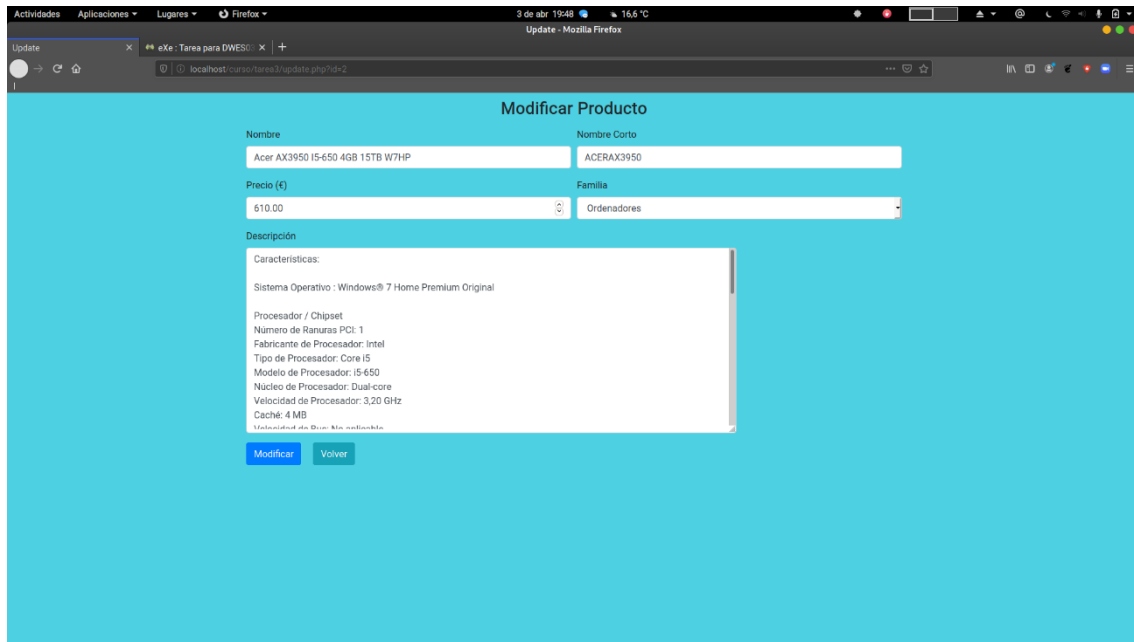
Código Familia: IMPRES

PVP (€): 99.90

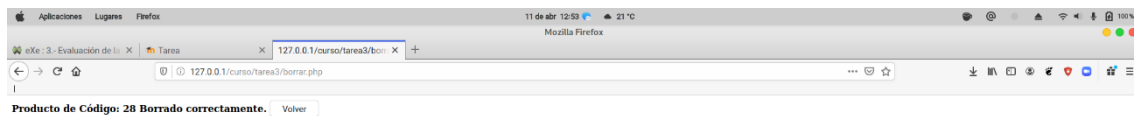
Descripción: Impresora laserjet P1102W es fácil de instalar y usar, además de que te ayudará a ahorrar energía y recursos. Olvidate de los cables y disfruta de la libertad que te proporciona su tecnología WIFI, imprime fácilmente desde cualquier de tu oficina. Formato máximo aceptado A4 A2 No A3 No A4 Si A5 Si A6 Si B5 Si B6 Si Sobres C5 (162 x 229 mm) Si Sobres C6 (114 x 162 mm) Si

[Volver](#)

- update.php. Nos aparecerá un formulario con los campos rellenos con los valores del producto seleccionado desde "listado.php" incluido el select donde seleccionamos la familia



- `borrar.php`. Será una página php con el código necesario para borrar el producto seleccionado desde "`listado.php`" un mensaje de información y un botón volver para volver a "`listado.php`".



Para acceder a la base de datos se debe usar PDO. Controlaremos y mostraremos los posibles errores. Para los estilos se recomienda usar Bootstrap.

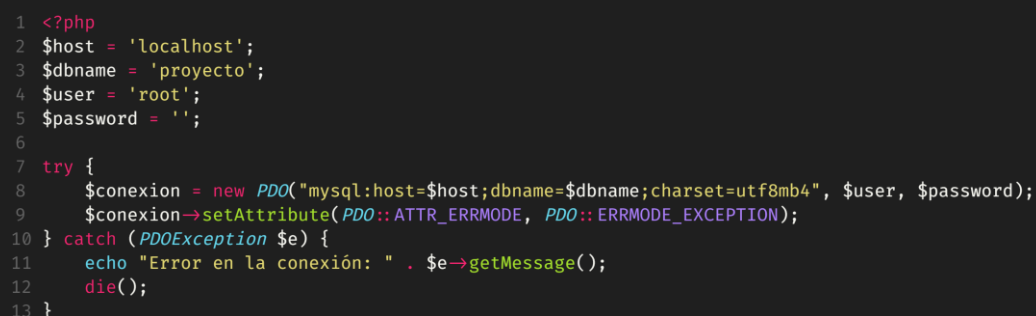
Pasaremos el código de producto por "get" tanto para "`detalle.php`" como para "`update.php`". Utilizando en el enlace "`detalle.php?id=cod`". En ambas páginas comprobaremos que esta variable existe, si no redireccionaremos a "`listado.php`" para esto podemos usar "`header('Location: listado.php')`".

# Explicación del código

## conexion.php

Este código es el encargado de conectarse a una base de datos MySQL, y para ello utiliza PDO (PHP Data Objects). PDO es la forma recomendada para trabajar con bases de datos en PHP, ya que es compatible con varios sistemas de gestión de bases de datos SGBD, lo que facilita el cambio en el futuro si fuera necesario.

Todo el código de conexion.php:



```
1 <?php
2 $host = 'localhost';
3 $dbname = 'proyecto';
4 $user = 'root';
5 $password = '';
6
7 try {
8     $conexion = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8mb4", $user, $password);
9     $conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
10 } catch (PDOException $e) {
11     echo "Error en la conexión: " . $e->getMessage();
12     die();
13 }
```

Datos de conexión:



```
1 $host = 'localhost';
2 $dbname = 'proyecto';
3 $user = 'root';
4 $password = '';
```

En este fragmento de código se definen las variables que contendrán la información necesaria para conectarse con la base de datos.

- \$host: contiene el servidor de la base de datos, en este caso es la máquina local, es decir, localhost.
- \$dbname: contiene el nombre de la base de datos, en este caso: proyecto.
- \$user: contiene el nombre del usuario, en este caso root, que es el usuario por defecto en MySQL.
- \$password: contiene la contraseña del usuario root. En este caso es '', lo que significa que no tiene contraseña.

A continuación, se crea un bloque try catch, esto se hace para poder capturar errores y excepciones durante la ejecución del código, lo que está dentro del bloque try es el código que se intenta (try) ejecutar, si por alguna razón ocurre algún error, se lanza una excepción que es capturada por el bloque catch. Lo que se logra con esto es evitar una interrupción brusca en la ejecución del programa para poder así averiguar que es lo que ha ocurrido.

```

1 try {
2     $conexion = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8mb4", $user, $password);
3     $conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

```

En primer lugar, se crea la conexión con la base de datos, para ello se le transfiere la información almacenada en las variables \$host, \$dbname, \$user, \$password, y además el conjunto de caracteres utilizados: utf8mb4.

En la segunda línea puede verse como se utiliza setAttribute, que es un atributo para la conexión que lanzará excepciones en caso de que ocurran errores, un ejemplo claro de excepción que se pretende capturar con esto es un error en la conexión.

```

1 } catch (PDOException $e) {
2     echo "Error en la conexión: " . $e->getMessage();
3     die();
4 }

```

PDOException es un tipo de excepción de conexión de bases de datos utilizando PDO, si fallara, se produciría una excepción de este tipo y que sería almacenada en la variable \$e

La siguiente línea simplemente muestra un mensaje de error personalizado: "Error en la conexión: " y luego lo concatenaría con . al mensaje almacenado en \$e. Así se sabría la razón del fallo.

Die() es una función que detiene la ejecución del programa, ya que no tiene sentido continuar si no se ha logrado la conexión con la base de datos.

## listado.php

```

1 <?php
2 require_once 'conexion.php';
3
4 $sql = "SELECT id, nombre, codigo FROM productos ORDER BY nombre ASC";
5 $stmt = $conexion->prepare($sql);
6 $stmt->execute();
7 $productos = $stmt->fetchAll(PDO::FETCH_ASSOC);
8 ?>

```

Con `require_once` se logra tener acceso a la variable `$conexion` del archivo `conexión.php` sin tener que volver a escribir el código. Es `require_once` para que el archivo `conexión.php` se incluya únicamente 1 vez, aunque se le llame varias veces en el código, lo que podría generar conflictos si llegar a ocurrir.

Al utilizar `"SELECT id, nombre, codigo FROM productos ORDER BY nombre ASC";`, se está solicitando las columnas `id`, `nombre` y `código` de la tabla `productos`, ordenados alfabéticamente de A a la Z.

`$stmt = $conexion->prepare($sql);` usamos el objeto `conexión` que fue creado en `conexión.php` y preparamos la consulta (con `prepare`) para evitar problemas de seguridad por inyección de sql. `$stmt` es una variable que lo que hace es representar la consulta preparada.

`$stmt->execute();` aquí ejecutamos la consulta que hemos preparado.

`$productos = $stmt->fetchAll(PDO::FETCH_ASSOC);` Aquí se recuperarán todos los resultados de la consulta SQL que se ejecutó anteriormente, `fetchAll` lo recuperará todo y `PDO::FETCH_ASSOC` hace que se devuelvan como un array asociativo en los que los nombres de las columnas se devolverán como claves de array.



```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Gestión de Productos</title>
7     <link rel="stylesheet" href="styles.css" />
8   </head>
9   <body>
10    <h1>Gestión de Productos</h1>
11
12    <section id="contenedor-principal-de-tabla">
13      <table>
14        <thead>
15          <tr>
16            <th>Detalle</th>
17            <th>ID</th>
18            <th>Código</th>
19            <th>Nombre</th>
20            <th>Acciones</th>
21          </tr>
22        </thead>
23        <tbody>
24          <?php foreach ($productos as $producto): ?>
25            <tr>
26              <td>
27                <form action="detalle.php" method="post" style="display:inline;">
28                  <input type="hidden" name="id" value="<?php echo htmlspecialchars($producto['id']); ?>">
29                  <button type="submit" class="detalle">Detalle</button>
30                </form>
31              </td>
32
33              <td><?php echo htmlspecialchars($producto['id']); ?></td>
34              <td><?php echo htmlspecialchars($producto['codigo']); ?></td>
35              <td><?php echo htmlspecialchars($producto['nombre']); ?></td>
36              <td>
37                <form action="update.php" method="post" style="display:inline;">
38                  <input type="hidden" name="id" value="<?php echo htmlspecialchars($producto['id']); ?>">
39                  <button type="submit" class="actualizar">Actualizar</button>
40                </form>
41
42                <form action="borrar.php" method="post" style="display:inline;">
43                  <input type="hidden" name="id" value="<?php echo htmlspecialchars($producto['id']); ?>">
44                  <button type="submit" class="borrar">Borrar</button>
45                </form>
46              </td>
47            </tr>
48          <?php endforeach; ?>
49        </tbody>
50      </table>
51      <button
52        type="button"
53        class="crear"
54        onclick="window.location.href='crear.php'"
55      >
56        Crear
57      </button>
58    </section>
59  </body>
60 </html>

```

En este código HTML no hay gran cosa que comentar realmente, todo se ha visto en las asignaturas anteriores del curso, y no merecen la pena ser explicados en esta asignatura que se centra en PHP.

Sin embargo, hay algunos pequeños puntos que sí merecen ser comentados:

<?php foreach (\$productos as \$producto): ?> Este es un bucle que generará una tabla dinámica, el bucle recorrerá la tabla de forma dinámica e irá añadiendo filas <tr> a la tabla.

```

1 <form action="detalle.php" method="post" style="display:inline;">
2     <input type="hidden" name="id" value="<?php echo htmlspecialchars($producto['id']); ?>">
3     <button type="submit" class="detalle">Detalle</button>
4 </form>

```

Se crea un formulario que enviará la ID del producto a detalle.php cuando el usuario pulse el botón detalle de la fila del producto indicado. El ID se enviará de forma oculta al enviar el formulario. Se utiliza `<?php echo htmlspecialchars($producto['id']); ?>` para evitar inyecciones HTML. Al presionar el botón se enviará el ID del producto a detalle.php para que muestre las características del producto que coincida con esa ID. **Todo este sistema explicado en este párrafo también será utilizado para los botones “Actualizar” y “Borrar”, con idénticos objetivos y funciones.**

```

1 <td><?php echo htmlspecialchars($producto['id']); ?></td>
2 <td><?php echo htmlspecialchars($producto['codigo']); ?></td>
3 <td><?php echo htmlspecialchars($producto['nombre']); ?></td>

```

Las líneas anteriores simplemente añaden los valores a la tabla dinámica y es la que en definitiva hace mostrar el listado completo de productos almacenados en la base de datos y que han sido capturados en la variable `$producto`.

`<?php endforeach; ?>` Este es el final del bucle y por tanto de la tabla dinámica.

## detalle.php

```

1 <?php
2 require_once 'conexion.php';
3
4 try {
5     $id = isset($_POST['id']) ? intval($_POST['id']) : 0;
6
7     $sql = "SELECT id, nombre, codigo, descripcion, precio, familia FROM productos WHERE id = :id";
8     $stmt = $conexion->prepare($sql);
9     $stmt->bindParam(':id', $id, PDO::PARAM_INT);
10    $stmt->execute();
11
12    $producto = $stmt->fetch(PDO::FETCH_ASSOC);
13
14    if (!$producto) {
15        echo "Producto no encontrado";
16        die();
17    }
18 } catch (PDOException $e) {
19     echo "Error en la conexión: " . $e->getMessage();
20     die();
21 }
22 ?>

```

`$id = isset($_POST['id']) ? intval($_POST['id']) : 0;` Aquí obtendremos el ID del producto que se envió desde listado.php mediante POST. Para lograr esto utilizaremos una expresión condicional ternaria, la primera parte verifica que se ha recibido la ID y que no es null, luego lo transforma en un número entero para asegurar que se trate a la ID como un número. Si la condición primera fuera falsa, es decir, no se ha enviado el ID, sería por defecto 0.

```
1 $sql = "SELECT id, nombre, codigo, descripcion, precio, familia FROM productos WHERE id = :id";
2 $stmt = $conexion->prepare($sql);
3 $stmt->bindParam(':id', $id, PDO::PARAM_INT);
4 $stmt->execute();
5
6 $producto = $stmt->fetch(PDO::FETCH_ASSOC);
```

En el código anterior se ve como se prepara la ejecución de la consulta SQL como se vió igualmente en listado.php. `$stmt->bindParam(':id', $id, PDO::PARAM_INT);` aquí se usa `bindParam` que es un método para enlazar un valor a un parámetro en una consulta SQL preparada. El parámetro `:id` es un marcador de posición, es un espacio reservado en el que se asignará el valor de `$id` usando `bindParam()`, por explicarlo mas brevemente, `:id` es el ID del producto enviado usando el formulario desde listado.php. `PDO::PARAM_INT` le dice al PDO que el valor debe ser tratado como un entero (y no como un string por ejemplo).

```
1 if (!$producto) {
2     echo "Producto no encontrado";
3     die();
4 }
```

Se verifica si el ID enviado existe en la base de datos, si fuera el caso, se mostraría el mensaje de error y se terminaría con la ejecución.

```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Detalle Producto</title>
7     <link rel="stylesheet" href="styles.css" />
8   </head>
9   <body>
10    <h1>Detalle Producto</h1>
11
12    <section class="detalle-cuadro">
13      <div class="detalle-cabecera"><?php echo htmlspecialchars($producto['nombre']); ?></div>
14      <div class="detalle-cuerpo">
15        <p class="centrar-texto"><strong>Código:</strong> <?php echo htmlspecialchars($producto['id']); ?></p>
16        <p><strong>Nombre:</strong> <?php echo htmlspecialchars($producto['nombre']); ?></p>
17        <p><strong>Nombre corto:</strong> <?php echo htmlspecialchars($producto['codigo']); ?></p>
18        <p><strong>Código Familia:</strong> <?php echo htmlspecialchars($producto['familia']); ?></p>
19        <p><strong>PVP (€):</strong> <?php echo htmlspecialchars($producto['precio']); ?></p>
20        <p><strong>Descripción:</strong> <?php echo htmlspecialchars($producto['descripcion']); ?></p>
21      </div>
22    </section>
23
24    <button
25      type="button"
26      class="btn-volver"
27      onclick="window.location.href='listado.php'"
28    >
29      Volver
30    </button>
31  </body>
32 </html>

```

Lo visto en el HTML anterior es lo mismo que se explicó en el apartado HTML de listado.php

## crear.php

```
1 <?php
2 require_once 'conexion.php';
3
4 try {
5     $sql_familias = "SELECT cod, nombre FROM familias";
6     $stmt_familias = $conexion->prepare($sql_familias);
7     $stmt_familias->execute();
8     $familias = $stmt_familias->fetchAll(PDO::FETCH_ASSOC);
9
10    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
11        $nombre = $_POST['nombre'] ?? '';
12        $precio = $_POST['precio'] ?? '';
13        $codigo = $_POST['nombre_corto'] ?? '';
14        $familia = $_POST['familia'] ?? '';
15        $descripcion = $_POST['descripcion'] ?? '';
16
17        if ($nombre && $precio && $codigo && $familia && $descripcion) {
18            $sql_last_id = "SELECT MAX(id) AS ultimo_id FROM productos";
19            $stmt_last_id = $conexion->prepare($sql_last_id);
20            $stmt_last_id->execute();
21            $ultimo_id = $stmt_last_id->fetch(PDO::FETCH_ASSOC)['ultimo_id'];
22            $nuevo_id = $ultimo_id + 1;
23
24            $sql_insert = "INSERT INTO productos (id, nombre, codigo, descripcion, precio, familia)
25                VALUES (:id, :nombre, :codigo, :descripcion, :precio, :familia)";
26            $stmt_insert = $conexion->prepare($sql_insert);
27            $stmt_insert->bindParam(':id', $nuevo_id);
28            $stmt_insert->bindParam(':nombre', $nombre);
29            $stmt_insert->bindParam(':codigo', $codigo);
30            $stmt_insert->bindParam(':descripcion', $descripcion);
31            $stmt_insert->bindParam(':precio', $precio);
32            $stmt_insert->bindParam(':familia', $familia);
33
34            if ($stmt_insert->execute()) {
35                echo "Producto creado con éxito.";
36            } else {
37                echo "Error al crear el producto.";
38            }
39        } else {
40            echo "Por favor, complete todos los campos.";
41        }
42    }
43 } catch (PDOException $e) {
44     echo "Error en la conexión: " . $e->getMessage();
45     die();
46 }
47 ?>
```

```
1 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
2     $nombre = $_POST['nombre'] ?? '';
3     $precio = $_POST['precio'] ?? '';
4     $codigo = $_POST['nombre_corto'] ?? '';
5     $familia = $_POST['familia'] ?? '';
6     $descripcion = $_POST['descripcion'] ?? '';
```

Aquí se verifica si el formulario ha sido enviado utilizando el método POST.

Se recuperan los datos enviados por el formulario, y se les asignan a las diferentes variables, si por alguna razón, alguno de los valores no están presentes, se le asignará mediante el operador ?? el valor "".

```
1 if ($nombre && $precio && $codigo && $familia && $descripcion) {  
2     $sql_last_id = "SELECT MAX(id) AS ultimo_id FROM productos";  
3     $stmt_last_id = $conexion->prepare($sql_last_id);  
4     $stmt_last_id->execute();  
5     $ultimo_id = $stmt_last_id->fetch(PDO::FETCH_ASSOC)['ultimo_id'];  
6     $nuevo_id = $ultimo_id + 1;
```

Se realiza una verificación para confirmar que todos los valores están presentes, si no es así, arrojará un error: echo "Por favor, complete todos los campos."; Luego se localiza el último ID de la tabla, es decir, el de mayor valor, y esto se utilizará para obtener un nuevo último ID que será asignado al nuevo producto que se está creando, así nos aseguraremos de que no se utilicen IDs ya en uso por otros productos por la cantidad de errores que esto provocaría al identificarlos.

```
1 $sql_insert = "INSERT INTO productos (id, nombre, codigo, descripcion, precio, familia)  
2     VALUES (:id, :nombre, :codigo, :descripcion, :precio, :familia)";  
3 $stmt_insert = $conexion->prepare($sql_insert);  
4 $stmt_insert->bindParam(':id', $nuevo_id);  
5 $stmt_insert->bindParam(':nombre', $nombre);  
6 $stmt_insert->bindParam(':codigo', $codigo);  
7 $stmt_insert->bindParam(':descripcion', $descripcion);  
8 $stmt_insert->bindParam(':precio', $precio);  
9 $stmt_insert->bindParam(':familia', $familia);
```

Se insertan los datos del nuevo producto que se desea crear en la base de datos.

```
1 if ($stmt_insert->execute()) {  
2     echo "Producto creado con éxito.";  
3 } else {  
4     echo "Error al crear el producto.";  
5 }
```

Ejecutamos la inserción de datos propiamente dicha y mostraremos un mensaje para confirmar al usuario si se ha logrado con éxito o no.



```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Crear Producto</title>
7     <link rel="stylesheet" href="styles.css" />
8   </head>
9   <body>
10    <h1>Crear Producto</h1>
11    <form action="crear.php" method="post">
12      <section class="form-group">
13        <article class="form-column">
14          <label for="nombre">Nombre</label>
15          <input
16            type="text"
17            id="nombre"
18            name="nombre"
19            placeholder="Nombre"
20            required
21            minlength="8"
22            pattern="[a-zA-Z0-9\s]+"
23            title="Debe tener al menos 8 caracteres alfanuméricos"
24          />
25
26          <label for="precio">Precio (€)</label>
27          <input
28            type="number"
29            id="precio"
30            name="precio"
31            placeholder="Precio (€)"
32            required
33            min="0"
34            step="any"
35            title="Debe introducir solo números"
36          />
37        </article>
38
39        <article class="form-column">
40          <label for="nombre_corto">Nombre corto</label>
41          <input
42            type="text"
43            id="nombre_corto"
44            name="nombre_corto"
45            placeholder="Nombre corto"
46            required
47            pattern="[a-zA-Z0-9\s]+"
48            title="Debe tener al menos 8 caracteres alfanuméricos"
49          />
50
51          <label for="familia">Familia</label>
52          <select id="familia" name="familia" required>
53            <option value="">Seleccione una opción</option>
54            <?php foreach ($familias as $familia): ?>
55              <option value="<?php echo htmlspecialchars($familia['cod']); ?>"
56                <?php echo htmlspecialchars($familia['nombre']); ?>
57            </option>
58            <?php endforeach; ?>
59          </select>
60        </article>
61      </section>
62
63      <label for="descripcion">Descripción</label>
64      <textarea
65        id="descripcion"
66        name="descripcion"
67        placeholder="Escriba aquí la descripción del producto..."
68        required
69        minlength="5"
70        title="Debe tener al menos 8 caracteres alfanuméricos"
71      ></textarea>
72
73      <section class="buttons">
74        <button type="submit" class="btn-crear">Crear</button>
75        <button type="reset" class="btn-limpiar">Limpiar</button>
76        <button
77          type="button"
78          class="btn-volver"
79          onclick="window.location.href='listado.php'"
80        >
81          Volver
82        </button>
83      </section>
84    </form>
85  </body>
86 </html>

```



```

1 <option value="">Seleccione una opción</option>
2 <?php foreach ($familias as $familia): ?>
3     <option value="<?php echo htmlspecialchars($familia['cod']); ?>">
4         <?php echo htmlspecialchars($familia['nombre']); ?>
5     </option>
6 <?php endforeach; ?>

```

En el HTML se insertarán las opciones para el selector del formulario directamente de las opciones disponibles en la tabla familia en la base de datos, usando un bucle foreach.

Como puede verse, se utiliza una vez más htmlspecialchars para evitar inyecciones maliciosas, al convertir los caracteres especiales en entidades HTML, esto significa que si introducen por ejemplo > o &, no lo interpretará así, sino como: &gt; o &amp; para que no sean interpretados como código HTML de la página y genere errores.

## update.php

```

1 <?php
2 require_once 'conexion.php';
3
4 try {
5     $id = isset($_POST['id']) ? intval($_POST['id']) : (isset($_GET['id']) ? intval($_GET['id']) : 0);
6
7     if ($id > 0) {
8         $sql_producto = "SELECT * FROM productos WHERE id = :id";
9         $stmt_producto = $conexion->prepare($sql_producto);
10        $stmt_producto->bindParam(':id', $id, PDO::PARAM_INT);
11        $stmt_producto->execute();
12        $producto = $stmt_producto->fetch(PDO::FETCH_ASSOC);
13
14        if (!$producto) {
15            echo "Producto no encontrado.";
16            die();
17        }
18
19        $sql_familias = "SELECT cod, nombre FROM familias";
20        $stmt_familias = $conexion->prepare($sql_familias);
21        $stmt_familias->execute();
22        $familias = $stmt_familias->fetchAll(PDO::FETCH_ASSOC);
23
24        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
25            $nombre = $_POST['nombre'] ?? '';
26            $precio = $_POST['precio'] ?? '';
27            $codigo = $_POST['nombre_corto'] ?? '';
28            $familia = $_POST['familia'] ?? '';
29            $descripcion = $_POST['descripcion'] ?? '';
30
31            if ($nombre && $precio && $codigo && $familia && $descripcion) {
32                $sql_update = "UPDATE productos SET nombre = :nombre, codigo = :codigo, descripcion = :descripcion, precio = :precio, familia = :familia WHERE id = :id";
33                $stmt_update = $conexion->prepare($sql_update);
34                $stmt_update->bindParam(':nombre', $nombre);
35                $stmt_update->bindParam(':codigo', $codigo);
36                $stmt_update->bindParam(':descripcion', $descripcion);
37                $stmt_update->bindParam(':precio', $precio);
38                $stmt_update->bindParam(':familia', $familia);
39                $stmt_update->bindParam(':id', $id, PDO::PARAM_INT);
40
41                if ($stmt_update->execute()) {
42                    echo "Producto modificado con éxito.";
43                } else {
44                    echo "Error al modificar el producto.";
45                }
46            } else {
47                echo "Por favor, complete todos los campos.";
48            }
49        } else {
50            echo "ID de producto inválido.";
51            die();
52        }
53    } catch (PDOException $e) {
54        echo "Error en la conexión: " . $e->getMessage();
55        die();
56    }
57 }
58 ?>

```

Verificamos si el ID fue enviado por el método POST, y si no, por el método GET (esto se debe a una modificación de última hora en el código debido a un requisito en el enunciado que pasó

por alto): \$id = isset(\$\_POST['id']) ? intval(\$\_POST['id']) : (isset(\$\_GET['id']) ? intval(\$\_GET['id']) : 0);

Se verifica el que el ID es igual o mayor a 0: if (\$id >= 0)

\$sql\_producto = "SELECT \* FROM productos WHERE id = :id"; se obtienen los datos que generen coincidencia en la base de datos con el ID proporcionado, ya que únicamente se cargarán los datos de ese producto en particular.

\$sql\_update = "UPDATE productos SET nombre = :nombre, codigo = :codigo, descripcion = :descripcion, precio = :precio, familia = :familia WHERE id = :id"; Con esto prepararemos la actualización de los datos en la base de datos, con la palabra clave SET indicaremos qué columnas queremos actualizar y con que valores se hará -> columna nombre con :nombre y así sucesivamente con todos los datos del producto.

```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Modificar Producto</title>
7     <link rel="stylesheet" href="styles.css" />
8   </head>
9   <body>
10    <h1>Modificar Producto</h1>
11    <form action="update.php?id=<?php echo $id; ?>" method="post">
12      <input type="hidden" name="id" value="<?php echo $id; ?>">
13      <section class="form-group">
14        <section class="form-column">
15          <label for="nombre">Nombre</label>
16          <input
17            type="text"
18            id="nombre"
19            name="nombre"
20            value="<?php echo htmlspecialchars($producto['nombre']); ?>"
21            placeholder="Nombre"
22            required
23            minlength="8"
24            pattern="[a-zA-Z0-9\s]+"
25            title="Debe tener al menos 8 caracteres alfanuméricos"
26          />
27
28          <label for="precio">Precio (€)</label>
29          <input
30            type="number"
31            id="precio"
32            name="precio"
33            value="<?php echo htmlspecialchars($producto['precio']); ?>"
34            placeholder="Precio (€)"
35            required
36            min="0"
37            step="any"
38            title="Debe introducir solo números"
39          />
40        </section>
41
42        <section class="form-column">
43          <label for="nombre_corto">Nombre corto</label>
44          <input
45            type="text"
46            id="nombre_corto"
47            name="nombre_corto"
48            value="<?php echo htmlspecialchars($producto['codigo']); ?>"
49            placeholder="Nombre corto"
50            required
51            pattern="[a-zA-Z0-9\s]+"
52            title="Debe tener al menos 8 caracteres alfanuméricos"
53          />
54
55          <label for="familia">Familia</label>
56          <select id="familia" name="familia" required>
57            <option value="">Seleccione una opción</option>
58            <?php foreach ($familias as $fam): ?>
59              <option value="<?php echo htmlspecialchars($fam['cod']); ?>"
60                <?php echo $fam['cod'] == $producto['familia'] ? 'selected' : '' ?>>
61                <?php echo htmlspecialchars($fam['nombre']); ?>
62              </option>
63            <?php endforeach; ?>
64          </select>
65        </section>
66      </section>
67
68      <label for="descripcion">Descripción</label>
69      <textarea
70        id="descripcion"
71        name="descripcion"
72        placeholder="Escriba aquí la descripción del producto..."
73        required
74        minlength="8"
75        title="Debe tener al menos 8 caracteres alfanuméricos"
76      ><?php echo htmlspecialchars($producto['descripcion']); ?></textarea>
77
78      <section class="buttons">
79        <button type="submit" class="btn-modificar">Modificar</button>
80        <button
81          type="button"
82          class="btn-volver"
83          onclick="window.location.href='listado.php'"
84        >
85          Volver
86        </button>
87      </section>
88    </form>
89  </body>
90 </html>

```

## borrar.php

```
1 <?php
2 require_once 'conexion.php';
3
4 if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['id'])) {
5     $id = intval($_POST['id']);
6
7     if ($id ≥ 0) {
8         $sql_delete = "DELETE FROM productos WHERE id = :id";
9         $stmt_delete = $conexion->prepare($sql_delete);
10        $stmt_delete->bindParam(':id', $id, PDO::PARAM_INT);
11
12        if ($stmt_delete->execute()) {
13            $mensaje = "Producto con ID: $id borrado correctamente.";
14        } else {
15            $mensaje = "Error al borrar el producto.";
16        }
17    } else {
18        $mensaje = "ID de producto inválido.";
19    }
20 } else {
21     $mensaje = "No se recibió un ID de producto.";
22 }
23 ?>
```

\$sql\_delete = "DELETE FROM productos WHERE id = :id"; Preparación del borrado de datos con la ID pasada a través de POST.

```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Borrar Producto</title>
7     <link rel="stylesheet" href="styles.css" />
8   </head>
9   <body id="fondo-blanco-seccion-borrar-producto" class="texto-a-la-izquierda">
10    <p class="mensaje-producto-borrado">
11      <?php echo $mensaje; ?>
12    </p>
13
14    <button
15      type="button"
16      class="btn-volver-seccion-borrar-producto"
17      onclick="window.location.href='listado.php'"
18    >
19      Volver
20    </button>
21  </body>
22 </html>

```

## Mostrando la ejecución del CRUD

### listado.php

Detalle	ID	Código	Nombre	Acciones
<a href="#">Detalle</a>	2	ACERAX3950	Acer AX3950 I5-650 4GB 1TB W7HP	<a href="#">Actualizar</a> <a href="#">Borrar</a>
<a href="#">Detalle</a>	3	ARCLPMP32GBN	Archos Clipper MP3 2GB negro	<a href="#">Actualizar</a> <a href="#">Borrar</a>
<a href="#">Detalle</a>	5	EEEEPC1005PXD	Asus EEEPC 1005PXD N455 1 250 BL	<a href="#">Actualizar</a> <a href="#">Borrar</a>
<a href="#">Detalle</a>	7	IXUS115HSAZ	Canon Ixus 115HS azul	<a href="#">Actualizar</a> <a href="#">Borrar</a>
<a href="#">Detalle</a>	11	LEGRIAFS306	Canon Legria FS306 plata	<a href="#">Actualizar</a> <a href="#">Borrar</a>
<a href="#">Detalle</a>	17	PIXMAIP4850	Canon Pixma IP4850	<a href="#">Actualizar</a> <a href="#">Borrar</a>
<a href="#">Detalle</a>	18	PIXMAMP252	Canon Pixma MP252	<a href="#">Actualizar</a> <a href="#">Borrar</a>
<a href="#">Detalle</a>	20	PWSHTA3100PT	Canon Powershot A3100 plata	<a href="#">Actualizar</a> <a href="#">Borrar</a>
<a href="#">Detalle</a>	26	ZENMP48GB300	Creative Zen MP4 8GB Style 300	<a href="#">Actualizar</a> <a href="#">Borrar</a>
<a href="#">Detalle</a>	24	STYLUSX515W	Epson Stylus SX515W	<a href="#">Actualizar</a> <a href="#">Borrar</a>

## crear.php

**Crear Producto**

**Nombre**  
Producto de prueba 1

**Nombre corto**  
PRUEBA01

**Precio (€)**  
2000

**Familia**  
Ordenadores

**Descripción**  
Esto es un ejemplo de descripción para el producto de prueba 1.

Crear Limpiar Volver

Se introducen unos valores para la creación de un nuevo producto y se pulsa en el botón “Crear”.

Producto creado con éxito.

**Crear Producto**

**Nombre**  
Nombre

**Nombre corto**  
Nombre corto

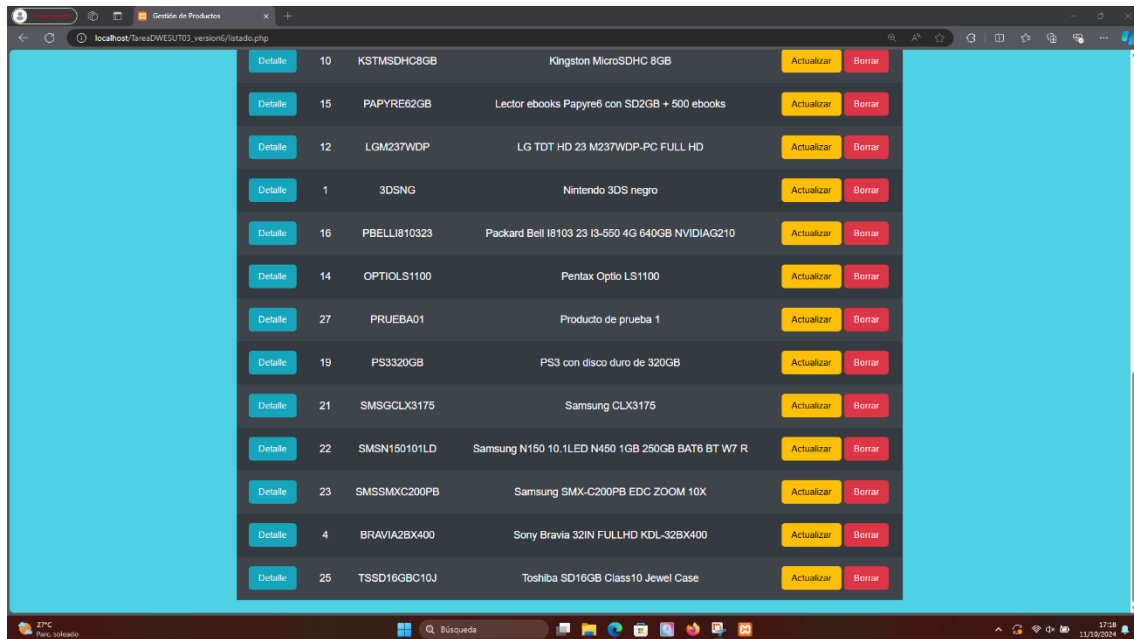
**Precio (€)**  
Precio (€)

**Familia**  
Seleccione una opción

**Descripción**  
Escriba aquí la descripción del producto...

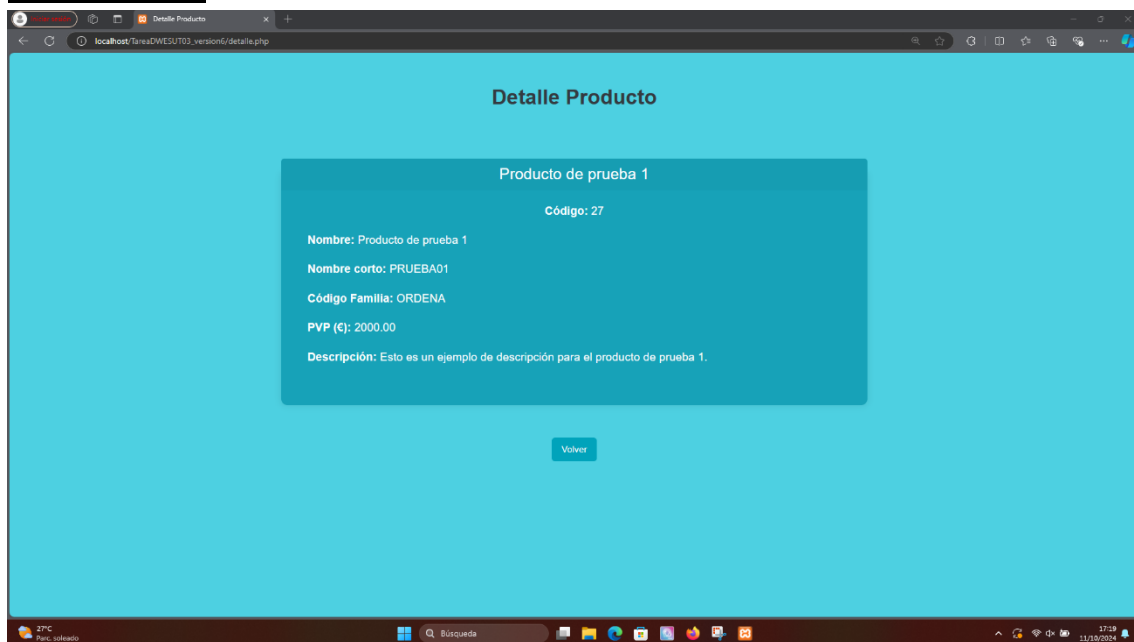
Crear Limpiar Volver

Como puede leerse en el mensaje de la parte superior, el producto ha sido creado con éxito.



Puede verse el producto creado en listado.php

## Detalle.php



Se abre y se muestran los detalles del producto nuevo creado en el paso anterior.

Por favor, complete todos los campos.

### Modificar Producto

<b>Nombre</b>	<b>Nombre corto</b>
<input type="text" value="Producto de prueba 1"/>	<input type="text" value="PRUEBA01"/>
<b>Precio (€)</b>	<b>Familia</b>
<input type="text" value="1000.00"/>	<input type="text" value="Ordenadores"/>
<b>Descripción</b>	
<input type="text" value="Esto es un ejemplo de modificación de la descripción para el producto de prueba 1."/>	

Se procede a modificar el producto creado anteriormente, cambiando su precio y modificando el texto de la descripción.

## update.php

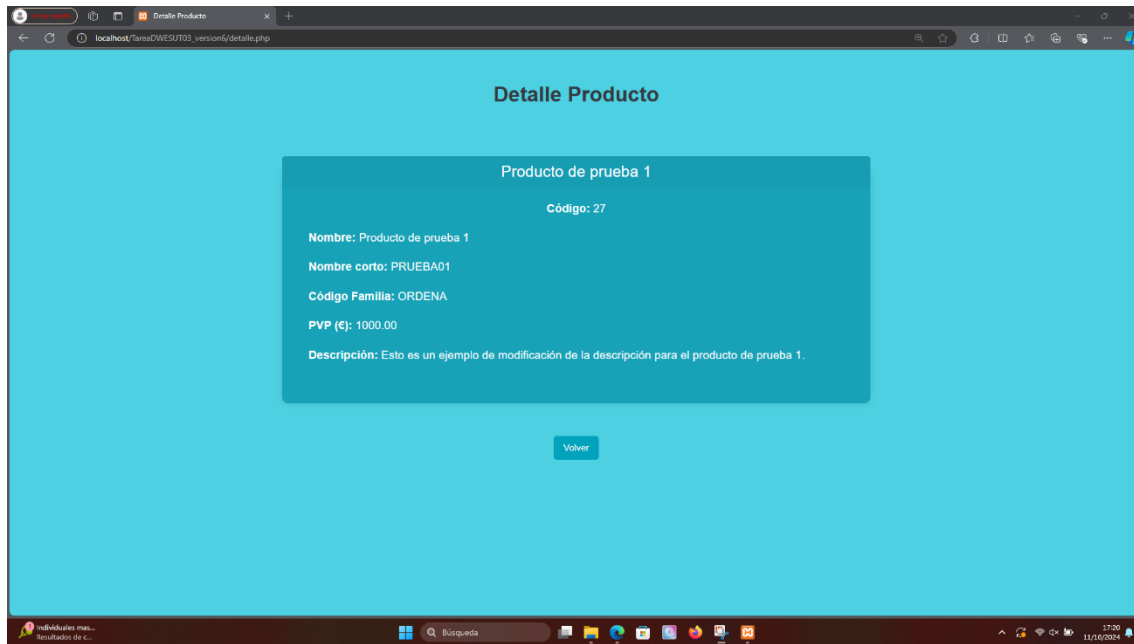
Producto modificado con éxito.

### Modificar Producto

<b>Nombre</b>	<b>Nombre corto</b>
<input type="text" value="Producto de prueba 1"/>	<input type="text" value="PRUEBA01"/>
<b>Precio (€)</b>	<b>Familia</b>
<input type="text" value="2000.00"/>	<input type="text" value="Ordenadores"/>
<b>Descripción</b>	
<input type="text" value="Esto es un ejemplo de descripción para el producto de prueba 1."/>	

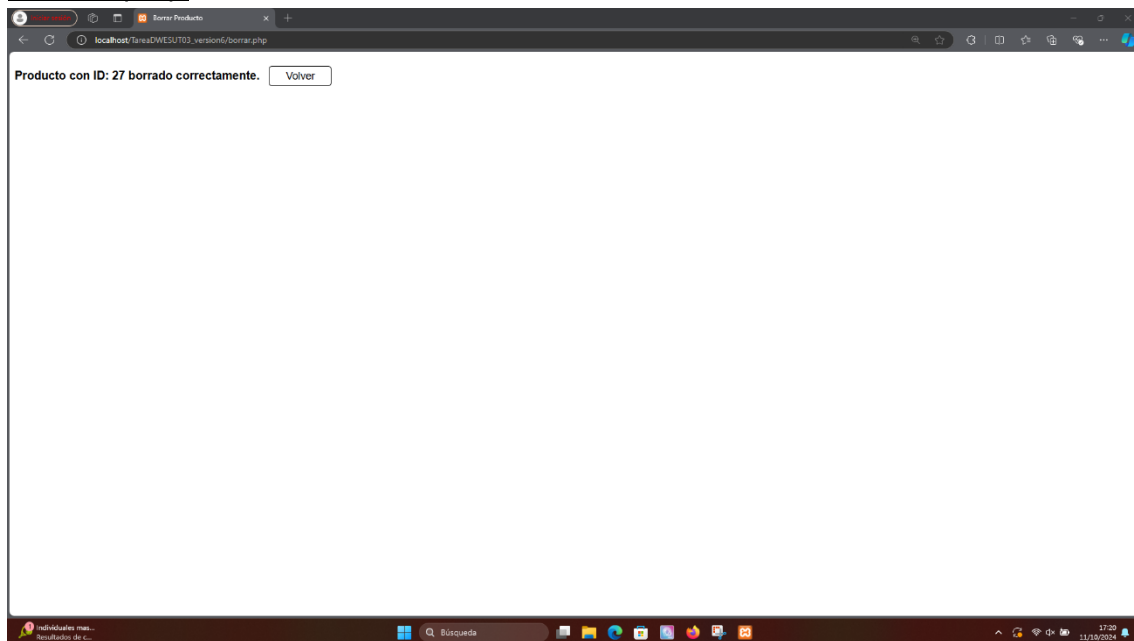
Como puede leerse en el mensaje de la parte superior, el producto ha sido modificado con éxito.





Puede verse en detalle.php que la modificación ha sido exitosa.

## borrar.php



Desde listado.php se pulsa el botón “Borrar” del producto que hemos creado y lleva a esta página con un mensaje sencillo indicando que ha sido correctamente borrado y cual es el ID del producto borrado.

**FIN**