

Tareas Unidad 7

Desarrollo Web en Entorno Servidor

Autor: Derimán Tejera Fumero

Fecha: 02/12/2024

Índice

Enunciado de la práctica	3
Explicación de la resolución de la práctica	5
Creación de la base de datos para la realización de la tarea.....	5
Archivos de la aplicación	5
Demostración del funcionamiento de la aplicación.....	11

Enunciado de la práctica

Implementaremos un sistema de votación en nuestra página de productos, de manera que, cualquier cliente validado pueda dar una puntuación del **1** al **5** a cada producto. Las valoraciones se reflejarán de manera inmediata en nuestra página gracias a **Xajax**.

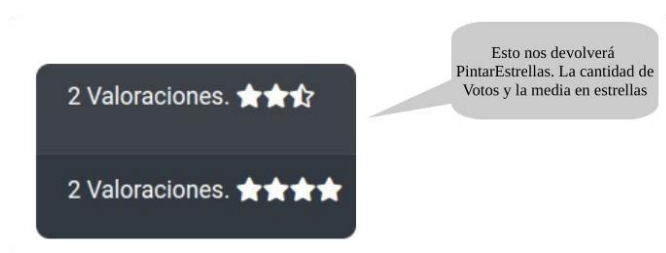
Haremos la página de Login similar a la del apartado **3.4**. En la validación, los errores se controlarán con **Xajax**.

Un cliente **NO** podrá valorar dos veces el mismo producto

Utilizaremos **Xajax** para presentar en tiempo real los cambios en la valoración cada vez que un cliente vote por un producto. Para ello implementaremos el método **PHP "miVoto"** que insertará el voto, si es la primera vez que el cliente valora un producto, y devolverá:

- La valoración de ese producto (la media de las valoraciones)
- False si el usuario ya ha valorado ese producto.

El método **PHP: "pintarEstrellas"** que se encargará de devolver el número de clientes que han valorado ese producto y las estrellas que se pintarán.



Para guardar los votos de cada producto por parte de cada cliente nos crearemos la tabla **"votos"** (se adjunta el guion **SQL** de la misma)

En la tabla usuarios, ya existente, guardaremos varios usuarios para ir votando por los productos. Se recomienda hacer uso de los iconos **Font Awesome** para representar las valoraciones. Fíjate en el vídeo

Las valoraciones de cada producto serán la media aritmética de las mismas, es decir, si un cliente ha valorado con **3/5** y otro con **5/5** la valoración será de **4**. **Como puntuamos sobre 5 la valoración media máxima será de 5 estrellas**. Si la parte decimal de la media de las valoraciones es superior o igual a 0.5 pintaremos media estrella.

Mostraremos también la cantidad de valoraciones que ha recibido un producto (es decir la cantidad de clientes que lo han votado).

Fíjate en el vídeo siguiente, podrás comprobar que están activadas las opciones de depuración en **Xajax**, lo que nos ayudará a detectar y corregir posibles errores de código.

En el **"onclick"** de botón botar llamaremos a la función **"JavaScript"** que se encargará de llamar a los métodos **PHP** ya mencionados

Guion con la tabla **"votos"** para añadir a la base de datos **"proyecto"**:

```

1 create table votos(
2     id int auto_increment primary key,
3     cantidad int default 0,
4     idPr int not null,
5     idUs varchar(20) not null,
6     constraint fk_votos_usu foreign key(idUs) references usuarios(usuario) on delete cascade on update cascade,
7     constraint fk_votos_pro foreign key(idPr) references productos(id) on delete cascade on update cascade
8 );

```

Actividades Aplicaciones Lugares Google Chrome 3 de may 19:54 29 °C

Formulario JAXON x +

127.0.0.1/cursos/tema7/practica7/login.php

Registro

usuario

contraseña

Registrar

0:01 / 1:30

Actividades Aplicaciones Lugares Google Chrome 3 de may 19:55 29 °C

Productos x +

127.0.0.1/cursos/tema7/practica7/listado.php

Productos onLine

inglés español Salir

Código	Nombre	Valoración	Valorar
2	Acer AX3950 i5-650 4GB 1TB W7HP 10mp	Sin valorar	1 <input type="button" value="Votar"/>
3	Archos Clipper MP3 64GB Negro	Sin valorar	1 <input type="button" value="Votar"/>
4	Sony Bravia 32IN FULLHD KDL-32BX400	Sin valorar	1 <input type="button" value="Votar"/>
5	Asus EEEPC 100SPXD N455 1 250 BL	Sin valorar	1 <input type="button" value="Votar"/>
6	HP Mini 110-3120 10.1LED N455 1GB 250GB W7S negro	Sin valorar	1 <input type="button" value="Votar"/>
7	Canon Ixus 115HS azul	Sin valorar	1 <input type="button" value="Votar"/>
8	Kingston DataTraveler 16GB DT101G2 USB2.0 negro	Sin valorar	1 <input type="button" value="Votar"/>
9	Kingston DataTraveler G3 32GB rojo	1 Valoraciones ★★☆☆	1 <input type="button" value="Votar"/>
10	Kingston MicroSDHC 8GB	Sin valorar	1 <input type="button" value="Votar"/>
11	Canon Legria FS306 Plata	1 Valoraciones ★★☆☆	1 <input type="button" value="Votar"/>
12	LG TDT HD 23 M237WDP-PC FULL HD	1 Valoraciones ★★☆☆	1 <input type="button" value="Votar"/>

0:18 / 1:30

Explicación de la resolución de la práctica

Creación de la base de datos para la realización de la tarea

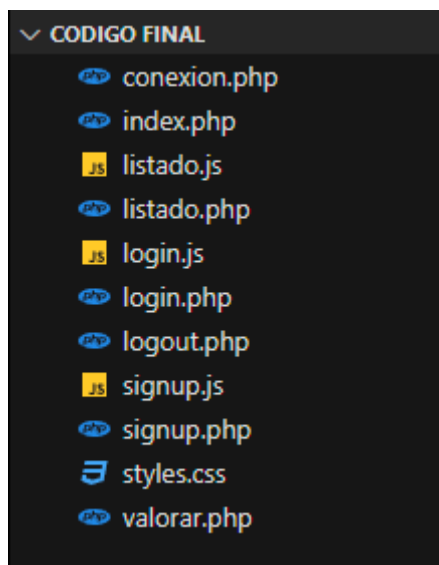
En primer lugar, partimos de la creación de la base de datos a la que llamaremos “tarea7”, para ello utilizaremos las instrucciones ya proporcionadas en tareas anteriores “proyecto.sql”, una vez importada esta base de datos, se añadirá el código indicado en el enunciado de la práctica para poder crear la “tabla votos”.

La estructura final de la base de datos la podrás encontrar en los archivos de código con el nombre: **“tarea7bd2 estructura base.sql”**. Esta base de datos está limpia de registros, pero es funcional.

La estructura final de la base de datos y además los datos utilizados en la prueba y demostración de esta práctica la podrás encontrar en: **“tarea7bd2 estructura base y datos de usuarios y valoraciones.sql”**.

Archivos de la aplicación

En primer lugar, partimos de los archivos que conforman la aplicación, únicamente se explicarán los archivos fundamentales de la tarea.



Conexión.php:

```

1 <?php
2 // Definimos las variables de conexión a la base de datos
3 $host = 'localhost';
4 $dbname = 'tarea7bd2';
5 $user = 'root';
6 $password = '';
7
8 try {
9     // Creamos una nueva conexión a la base de datos on PDO
10    $conexion = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8mb4", $user, $password);
11    // Configuramos el modo de error para lanzar excepciones si hay un fallo
12    $conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
13 } catch (PDOException $e) {
14     // Si ocurre un error en la conexión mostraremos un mensaje
15     echo "Error en la conexión: " . $e->getMessage();
16     die(); // Detenemos la ejecución del script
17 }

```

Index.php:

```

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>Bienvenido</title>
7     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
8     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.4/css/all.css">
9     <link rel="stylesheet" href="styles.css" />
10 </head>
11 <body>
12
13 <div class="container">
14     <h1 class="mb-4">Bienvenido</h1>
15     <a href="signup.php" class="btn btn-primary btn-lg">
16         <i class="fas fa-user-plus"></i> Registrarse
17     </a>
18     <a href="login.php" class="btn btn-success btn-lg">
19         <i class="fas fa-sign-in-alt"></i> Iniciar sesión
20     </a>
21 </div>
22
23 </body>
24 </html>

```

Esta web index servirá como introducción, no la pedía el ejercicio explícitamente pero por comodidad y uso para las pruebas le terminé creando. No hay gran cosa que explicar aquí porque no se usa PHP.

Login.php:

Para realizar el login, se ha modificado una página mostrada en el temario.

```

1 <?php
2 session_start();
3
4 require_once 'conexion.php';
5
6 if (isset($_SESSION['usuario'])) {
7     header("Location: listado.php");
8     exit();
9 }
10
11 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
12     $usuario = $_POST['usu'] ?? '';
13     $password = $_POST['pass1'] ?? '';
14
15     $response = ['success' => false, 'message' => ''];
16
17     if (empty($usuario) || empty($password)) {
18         $response['message'] = 'Usuario y contraseña son obligatorios.';
19     } else {
20         $passwordCifrada = sha1($password);
21
22         $stmt = $conexion->prepare("SELECT * FROM usuarios WHERE usuario = :usuario AND pass = :password");
23         $stmt->execute(['usuario' => $usuario, 'password' => $passwordCifrada]);
24         $user = $stmt->fetch(PDO::FETCH_ASSOC);
25
26         if ($user) {
27             $_SESSION['usuario'] = $usuario;
28             $response['success'] = true;
29             $response['redirect'] = 'listado.php';
30         } else {
31             $response['message'] = 'Usuario o contraseña incorrectos.';
32         }
33     }
34
35     header('Content-Type: application/json');
36     echo json_encode($response);
37     exit;
38 }
39 ?>
40
41 <!DOCTYPE html>
42 <html lang="es">
43 <head>
44     <meta charset="UTF-8">
45     <meta name="viewport" content="width=device-width, initial-scale=1.0">
46     <title>Iniciar Sesión</title>
47     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
48     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css">
49     <script src="login.js" defer></script>
50     <link rel="stylesheet" href="styles.css" />
51 </head>
52 <body>
53 <div class="container mt-5">
54     <div class="d-flex justify-content-center h-100">
55         <div class="card" style="width: 24rem;">
56             <div class="card-header">
57                 <h3><i class="fa fa-cog mr-1"></i>Iniciar sesión</h3>
58             </div>
59             <div class="card-body">
60                 <form id="miForm" method="POST" onsubmit="event.preventDefault(); enviarFormulario();">
61                     <div class="input-group form-group">
62                         <div class="input-group-prepend">
63                             <span class="input-group-text"><i class="fas fa-user"></i></span>
64                         </div>
65                         <input type="text" class="form-control" placeholder="Usuario" id="usu" name="usu" required>
66                     </div>
67                     <div class="input-group form-group">
68                         <div class="input-group-prepend">
69                             <span class="input-group-text"><i class="fas fa-key"></i></span>
70                         </div>
71                         <input type="password" class="form-control" placeholder="Contraseña" id="pass1" name="pass1" required>
72                     </div>
73                     <div class="botones-login-y-signup">
74                         <a href="index.php" class="btn btn-secondary">
75                             <i class="fas fa-arrow-left"></i> Volver
76                         </a>
77                         <input type="submit" value="Entrar" class="btn btn-info">
78                     </div>
79                 </form>
80             </div>
81         </div>
82     </div>
83 </div>
84 </body>
85 </html>

```

```

1 <?php
2 // Iniciamos la sesión para poder gestionar datos entre diferentes páginas
3 session_start();
4
5 // Incluimos el archivo de conexión a la base de datos
6 require_once 'conexion.php';
7
8 // Verificamos si ya existe una sesión activa para este usuario
9 if (isset($_SESSION['usuario'])) {
10     // Si el usuario ya inició sesión lo redirigimos al listado de productos
11     header("Location: listado.php");
12     exit();
13 }
14
15 // Comprobamos si el formulario fue enviado con el método POST
16 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
17     // Recogemos los datos del formulario con un valor por defecto vacío
18     $usuario = $_POST['usu'] ?? '';
19     $password = $_POST['pass1'] ?? '';
20
21     // Preparamos una respuesta inicial en formato JSON para el cliente
22     $response = ['success' => false, 'message' => ''];
23
24     // Validamos que los campos no estén vacíos
25     if (empty($usuario) || empty($password)) {
26         // Informamos al cliente que el usuario y la contraseña son obligatorios
27         $response['message'] = 'Usuario y contraseña son obligatorios.';
28     } else {
29         // Ciframos la contraseña ingresada usando sha1 para compararla con la base de datos.
30         $passwordCifrada = sha1($password);
31
32         // Preparamos la consulta SQL para buscar al usuario con las credenciales proporcionadas
33         $stmt = $conexion->prepare("SELECT * FROM usuarios WHERE usuario = :usuario AND pass = :password");
34         $stmt->execute(['usuario' => $usuario, 'password' => $passwordCifrada]);
35
36         // Recuperamos los datos del usuario si las credenciales son correctas
37         $user = $stmt->fetch(PDO::FETCH_ASSOC);
38
39         // Evaluamos si se encontró un usuario con esas credenciales
40         if ($user) {
41             // Guardamos el usuario en la sesión para recordar que ha iniciado sesión
42             $_SESSION['usuario'] = $usuario;
43             // Indicamos que todo ha ocurrido de forma exitosa y redireccionamos al listado.php
44             $response['success'] = true;
45             $response['redirect'] = 'listado.php';
46         } else {
47             // Si las credenciales son incorrectas, informamos al cliente
48             $response['message'] = 'Usuario o contraseña incorrectos.';
49         }
50     }
51
52     // Indicamos que la respuesta será en formato JSON
53     header('Content-Type: application/json');
54     // Enviamos la respuesta al cliente
55     echo json_encode($response);
56     // Finalizamos el script
57     exit;
58 }
59 ?>

```

Login.js:


```
1 // Creamos una función asíncrona para enviar el formulario de inicio de sesión
2 async function enviarFormulario() {
3   // Obtenemos el formulario mediante su ID
4   const form = document.getElementById("miForm");
5
6   // Creamos un objeto para recoger los datos del formulario
7   const formData = new FormData(form);
8
9   try {
10    // Realizamos una petición a login.php usando POST y enviando los datos del formulario
11    const response = await fetch("login.php", {
12      method: "POST",
13      body: formData,
14    });
15
16    // Convertimos la respuesta del servidor en formato JSON
17    const result = await response.json();
18
19    // Verificamos si el inicio de sesión fue correcto
20    if (result.success) {
21      // Redirigimos al usuario a la página indicada en la respuesta
22      window.location.href = result.redirect;
23    } else {
24      // Mostramos un mensaje de error si las credenciales son incorrectas
25      alert(result.message);
26    }
27  } catch (error) {
28    // Si ocurre un error en el servidor o en la comunicación mostraremos un mensaje al usuario
29    alert("Ocurrió un error en el servidor.");
30  }
31 }
```

Listado.php:

```

1 <?php
2 session_start(); // Iniciamos la sesión para mantener el estado del usuario entre las páginas
3 require_once 'conexion.php'; // Incluimos el archivo que establece la conexión con la base de datos
4
5 // Verificamos si el usuario ha iniciado sesión
6 if (!isset($_SESSION['usuario'])) {
7     // Si no hay sesión activa, redirigimos al usuario a la página de inicio de sesión
8     header("Location: login.php");
9     exit(); // Detenemos la ejecución del script para evitar cualquier otro procesamiento
10 }
11
12 // Obtenemos el nombre del usuario actual de la sesión
13 $usuarioActual = $_SESSION['usuario'];
14
15 // Consulta para obtener la lista de productos y sus valoraciones
16 $sql = "
17     SELECT
18         p.id,
19         p.nombre,
20         IFNULL(ROUND(AVG(v.cantidad), 1), 0) AS valoracion_media,
21         COUNT(v.cantidad) AS num_votos
22     FROM productos p
23     LEFT JOIN votos v ON p.id = v.idPr
24     GROUP BY p.id, p.nombre
25     ORDER BY p.id ASC";
26
27 $stmt = $conexion->prepare($sql); // Preparamos la consulta para evitar inyecciones SQL
28 $stmt->execute(); // Ejecutamos la consulta en la base de datos
29 $productos = $stmt->fetchAll(PDO::FETCH_ASSOC); // Obtenemos todos los resultados en un array asociativo
30
31 // Consulta para obtener las valoraciones realizadas por el usuario actual
32 $sqlVotosUsuario = "
33     SELECT idPr, cantidad
34     FROM votos
35     WHERE idUs = :usuario";
36
37 $stmtVotos = $conexion->prepare($sqlVotosUsuario); // Preparamos la consulta
38 $stmtVotos->execute(['usuario' => $usuarioActual]); // Ejecutamos la consulta, pasando el usuario actual como parámetro
39 $votosUsuario = $stmtVotos->fetchAll(PDO::FETCH_KEY_PAIR); // Obtenemos un array asociativo [idPr => cantidad]
40 ?>

```

Listado.js:

```

1 // Función asincrónica para gestionar la votación de un producto
2 async function votarProducto(idProducto, valoracion) {
3     try {
4         // Enviamos la solicitud al servidor mediante fetch api
5         const response = await fetch("valorar.php", {
6             method: "POST",
7             headers: { "Content-Type": "application/x-www-form-urlencoded" }, // Cabecera para especificar el tipo de contenido
8             body: new URLSearchParams({ id: idProducto, valoracion }) // Cuerpo de la solicitud con los datos del producto y la valoración
9         });
10
11         // Obtenemos la respuesta en JSON
12         const result = await response.json();
13
14         if (result.success) {
15             // Si la votación fue exitosa mostramos un mensaje y recargamos la página
16             alert(result.message);
17             location.reload(); // Recargamos la página para ver los cambios
18         } else {
19             // Si hubo un error en la votación mostramos el error
20             alert(result.message);
21         }
22     } catch (error) {
23         // Si ocurre un error técnico
24         alert("Error al realizar la votación: " + error.message);
25     }
26 }

```

Logout.php:

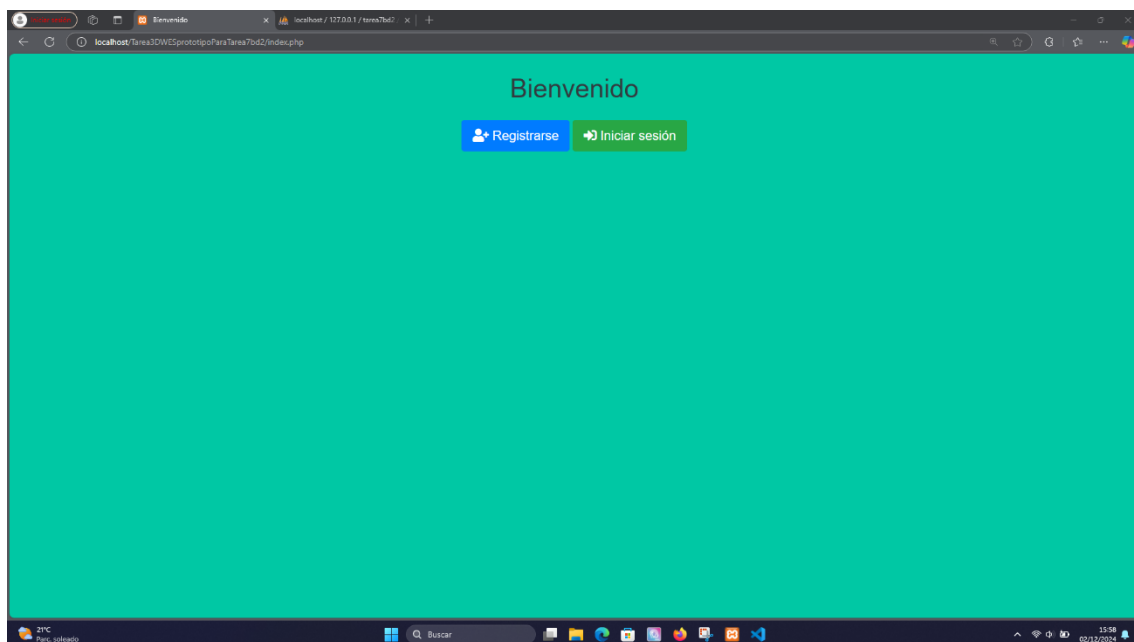
```

1 <?php
2 session_start(); // Iniciamos sesión para poder interactuar con la sesión que esté activa.
3
4 // Destruimos todas las sesiones activas
5 session_unset(); // Eliminamos todas las variables de la sesión activa
6 session_destroy(); // Destruimos completamente la sesión activa en el servidor (recomiendan hacerlo por seguridad)
7
8 header("Location: index.php"); // Redirigimos al usuario a la página al cerrar la sesión
9 exit(); // Finalizams el script
10 ?>

```

Demostración del funcionamiento de la aplicación

Página de inicio de la aplicación, aquí el usuario podrá seleccionar entre registrar un usuario nuevo o iniciar sesión con un usuario ya existente. Este apartado no se ha solicitado en el enunciado de la tarea, pero como no requería demasiado código, he decidido implementarlo a título personal para facilitar la demostración de la aplicación y la introducción de datos.

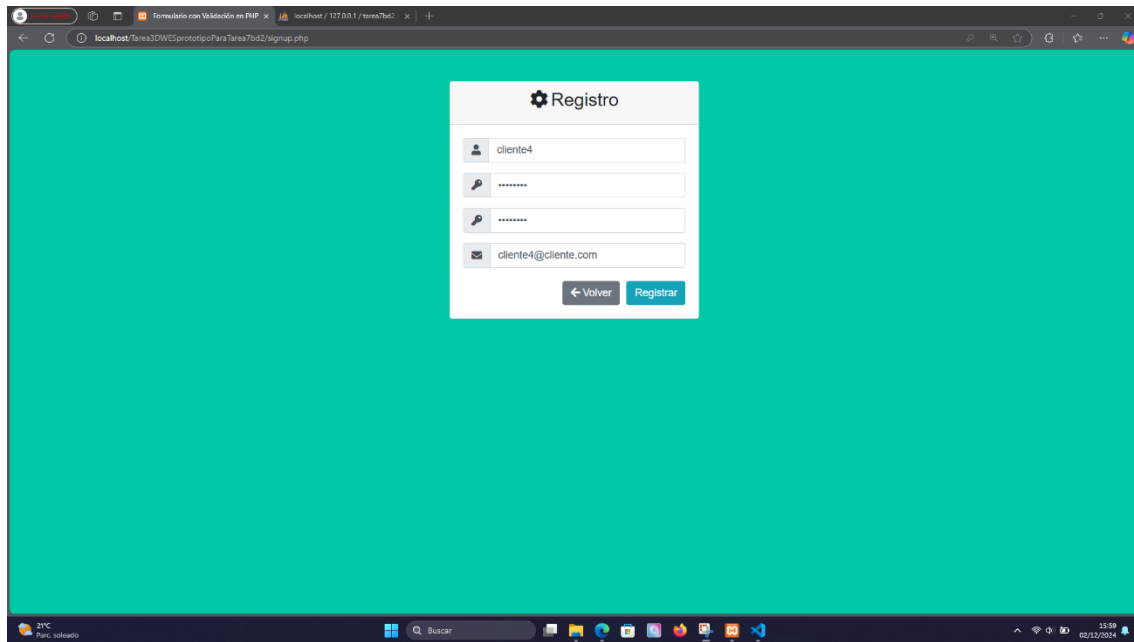


Si el usuario hace clic en registrar usuario se encontrará en signup.php que permite el registro de usuarios. Para este ejemplo se han utilizado los siguientes datos:

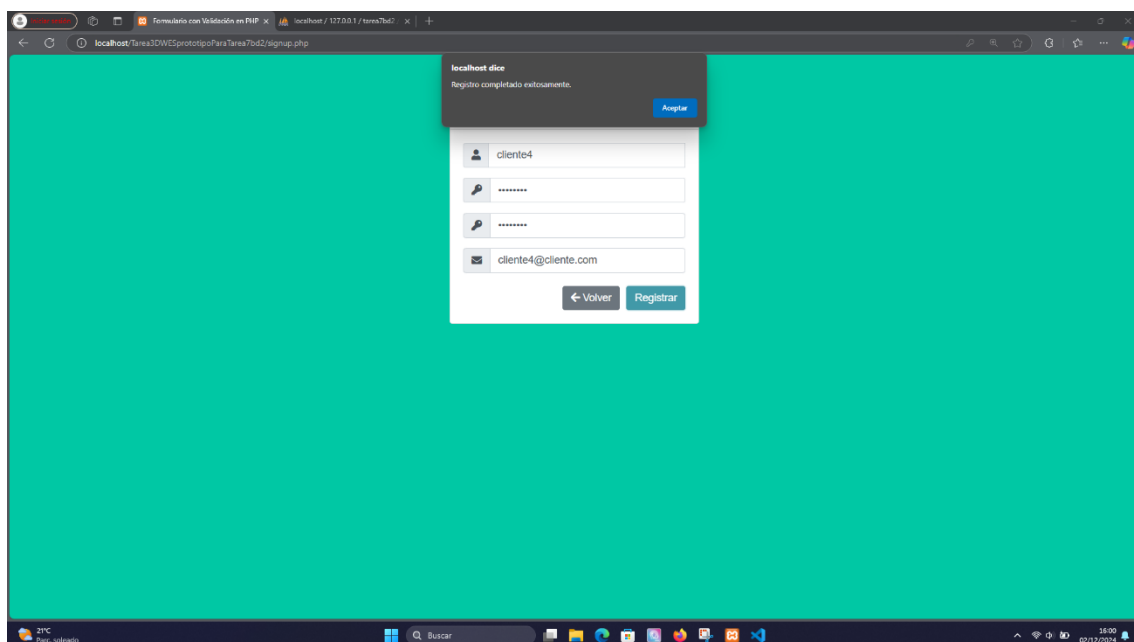
Nombre: cliente4

Contraseña: cliente4

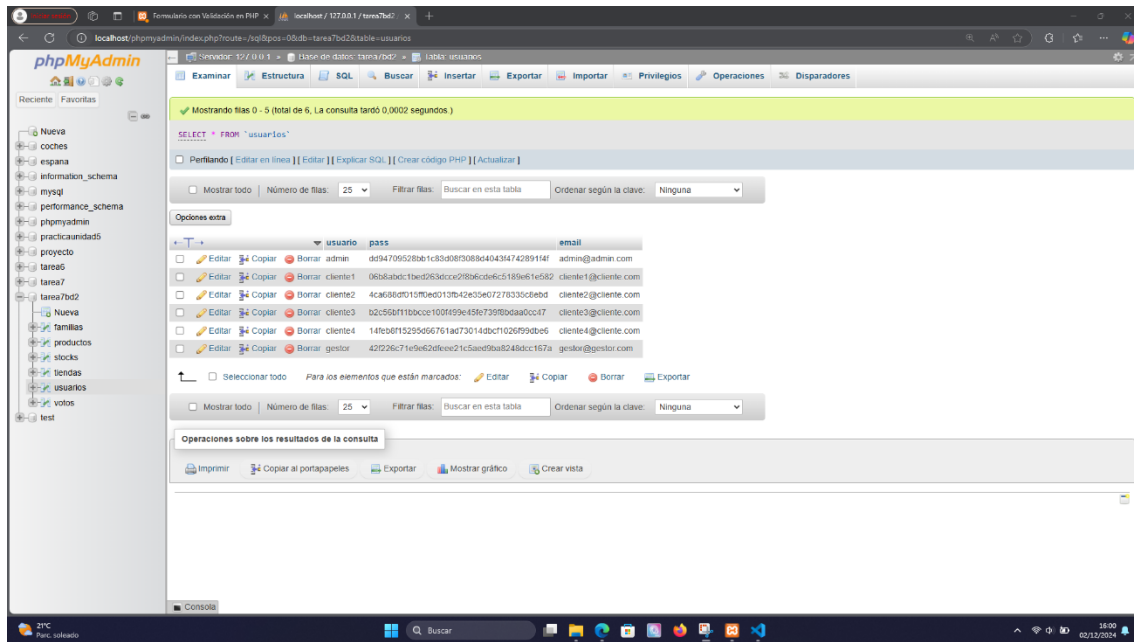
Email: cliente4@cliente.com



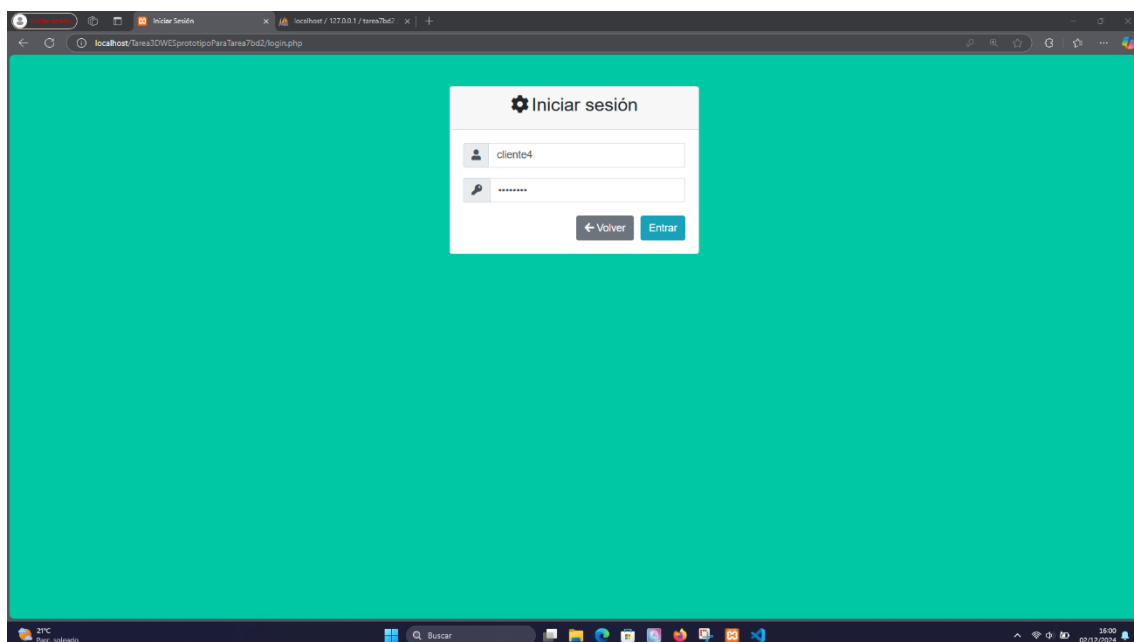
Como puede verse, el usuario que queremos introducir a superado todas las verificaciones y un mensaje en la parte superior indica que ha sido añadido satisfactoriamente a la base de datos.



En phpmyadmin, podemos ver como ha sido añadido el usuario cliente4 junto con el resto de usuario ya existentes en la tabla “usuarios”.

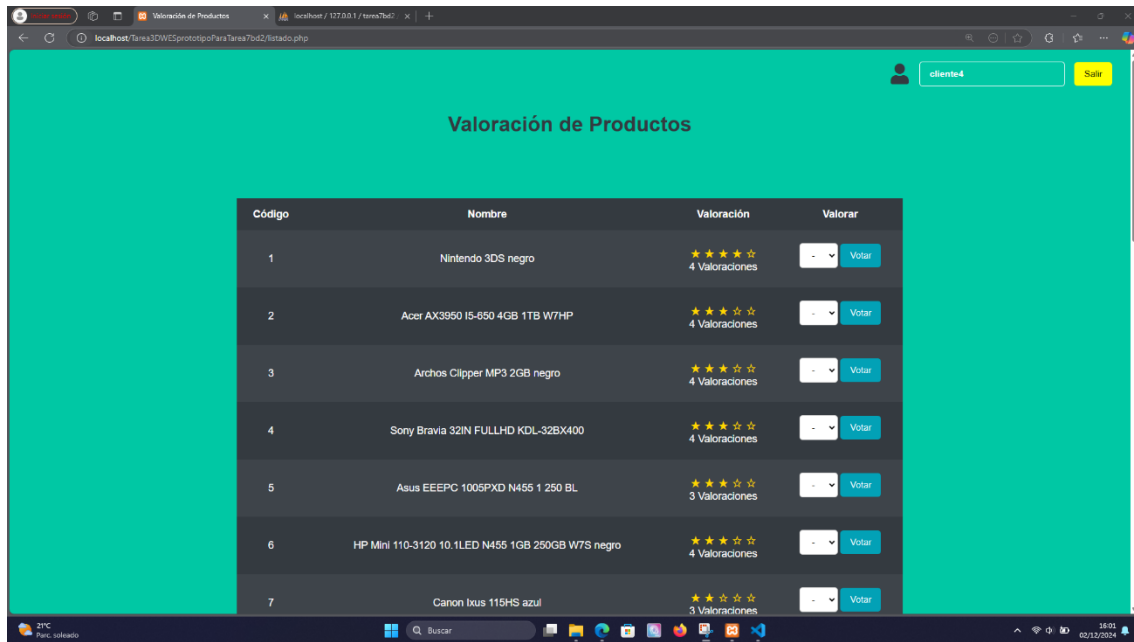


Ahora vamos a usar el usuario que hemos creado, para ello volvemos a index.php y seleccionamos “Iniciar sesión”, lo que veremos a continuación es esto:



Introducimos los datos de cliente4 para poder entrar.

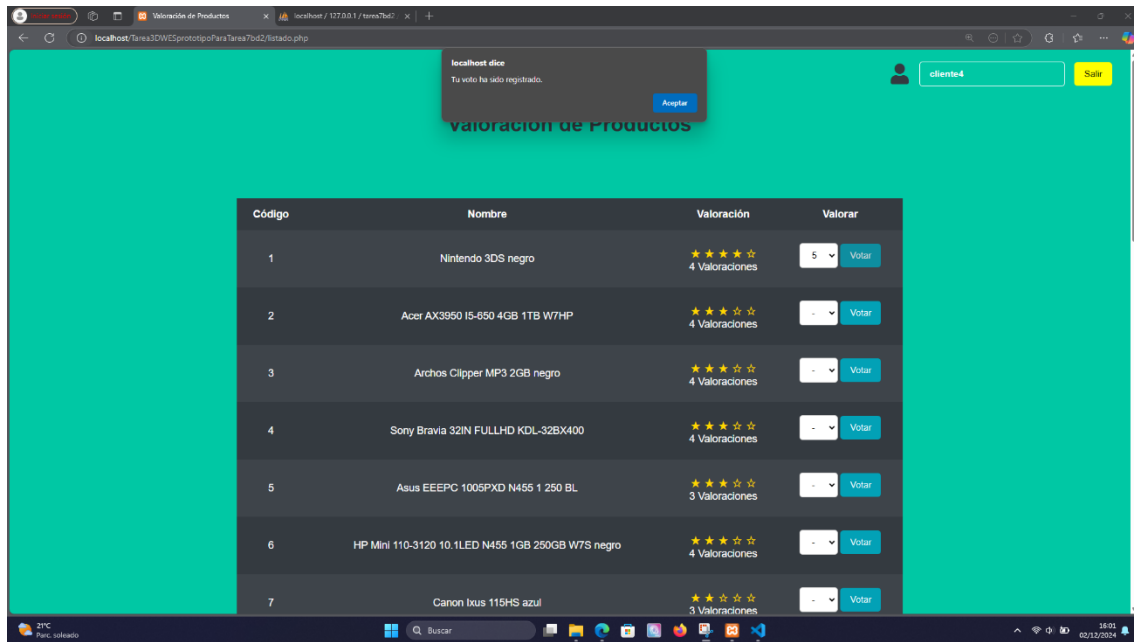
Ya hemos entrado en la sesión de cliente4, en la esquina superior derecha, junto al botón “Salir” de color amarillo, puede verse el nombre del propietario de la sesión. Aquí puede verse el listado de productos de la tienda de la aplicación y la valoración de otros usuarios de estos productos:



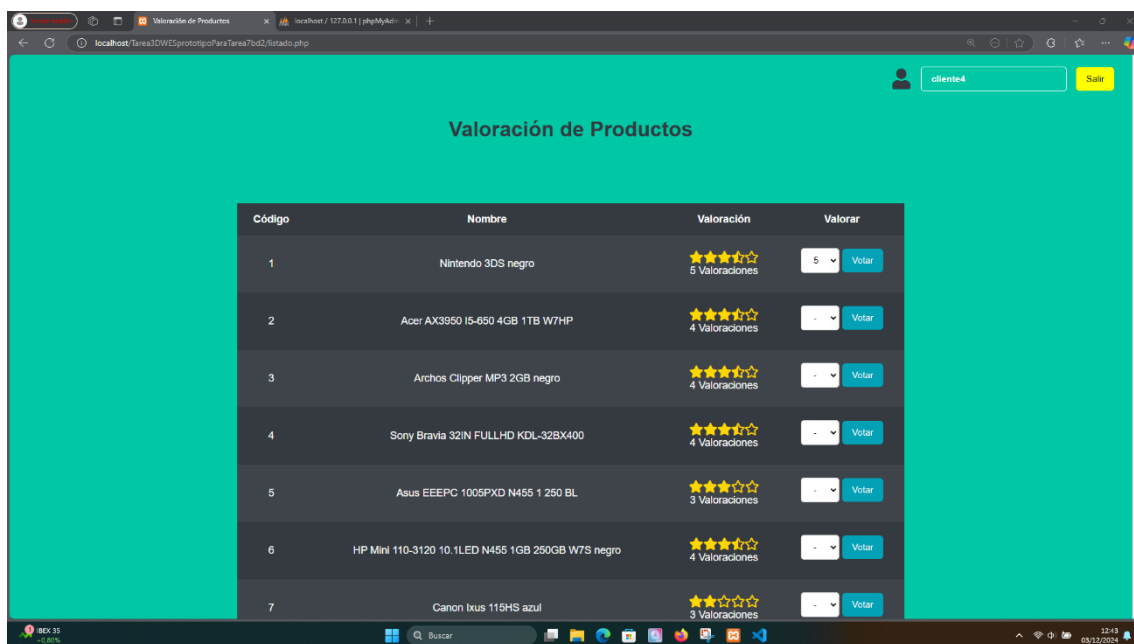
Para este ejemplo, cliente4 va a votar con 5 estrellas al producto “Nintendo 3DS negro”, como puede verse, se ha utilizado el selector para elegir el número 5 y luego se ha hecho clic en el botón “Votar”.



Un mensaje indica que el voto se ha enviado correctamente:

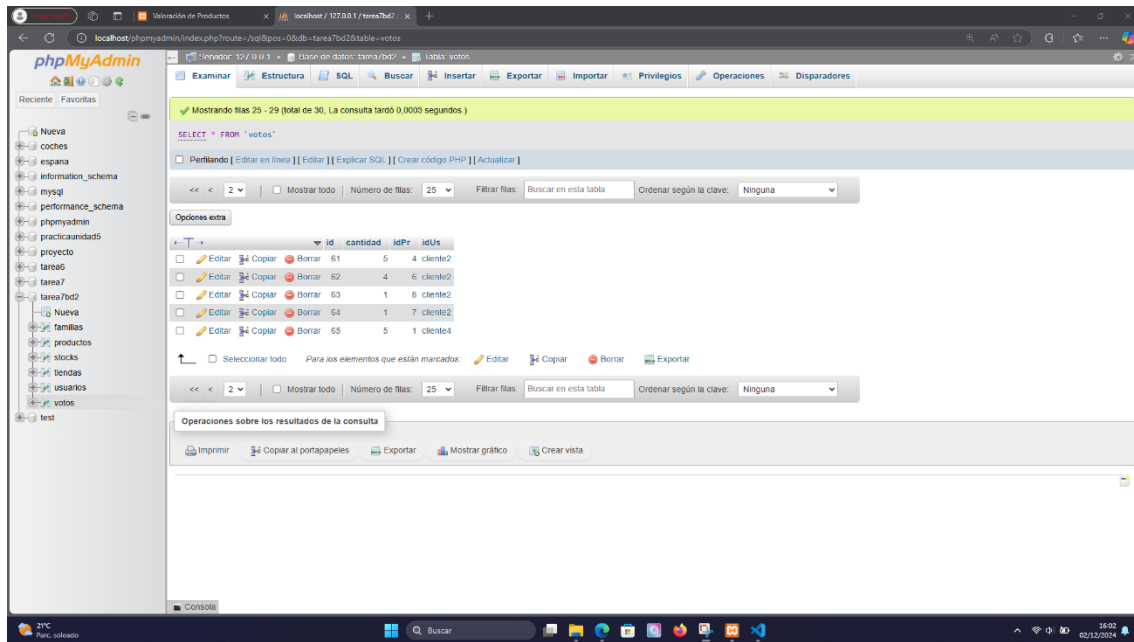


Ahora puede verse como se ha sumado el voto de cliente4 a los ya existentes anteriormente, ahora hay 5 valoraciones, en lugar de 4 como había antes:

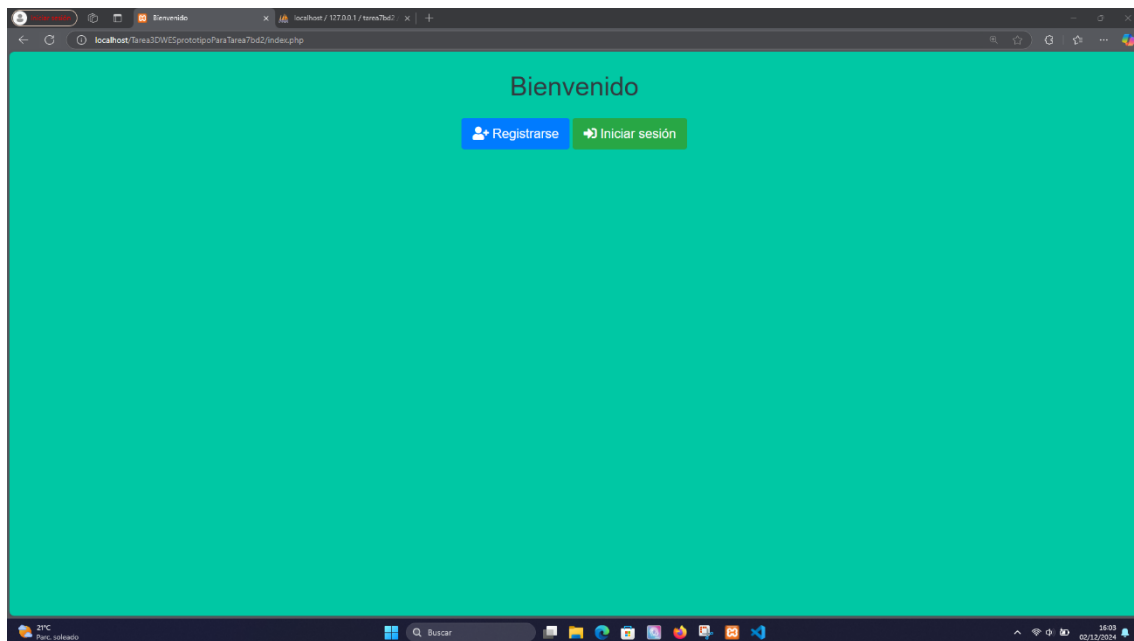


El usuario cliente4 solo podrá votar 1 vez un mismo producto, sin embargo, puede actualizar su voto sobre ese producto, para esto, es tan simple como elegir un nuevo número de estrellas y darle a “Votar” nuevamente.

En la tabla “votos” se almacenarán los votos sobre los productos y que usuario ha realizado el voto y qué ha votado, esto es esencial para evitar que un usuario vote mas de una vez por un producto y para poder modificar el voto sobre un producto sobre el que ya se ha votado:



Al presionar el botón amarillo de “Salir” de la esquina superior derecha, se saldrá de la sesión del usuario cliente4 y se redirigirá a index.php:



FIN