

# EL REFUGIO DEL LECTOR

## Proyecto grupal

Roberto Angello Horna Periche

Angely Villanueva Espinosa

2º - DAW

Octubre 2030

# EL REFUGIO DEL LECTOR

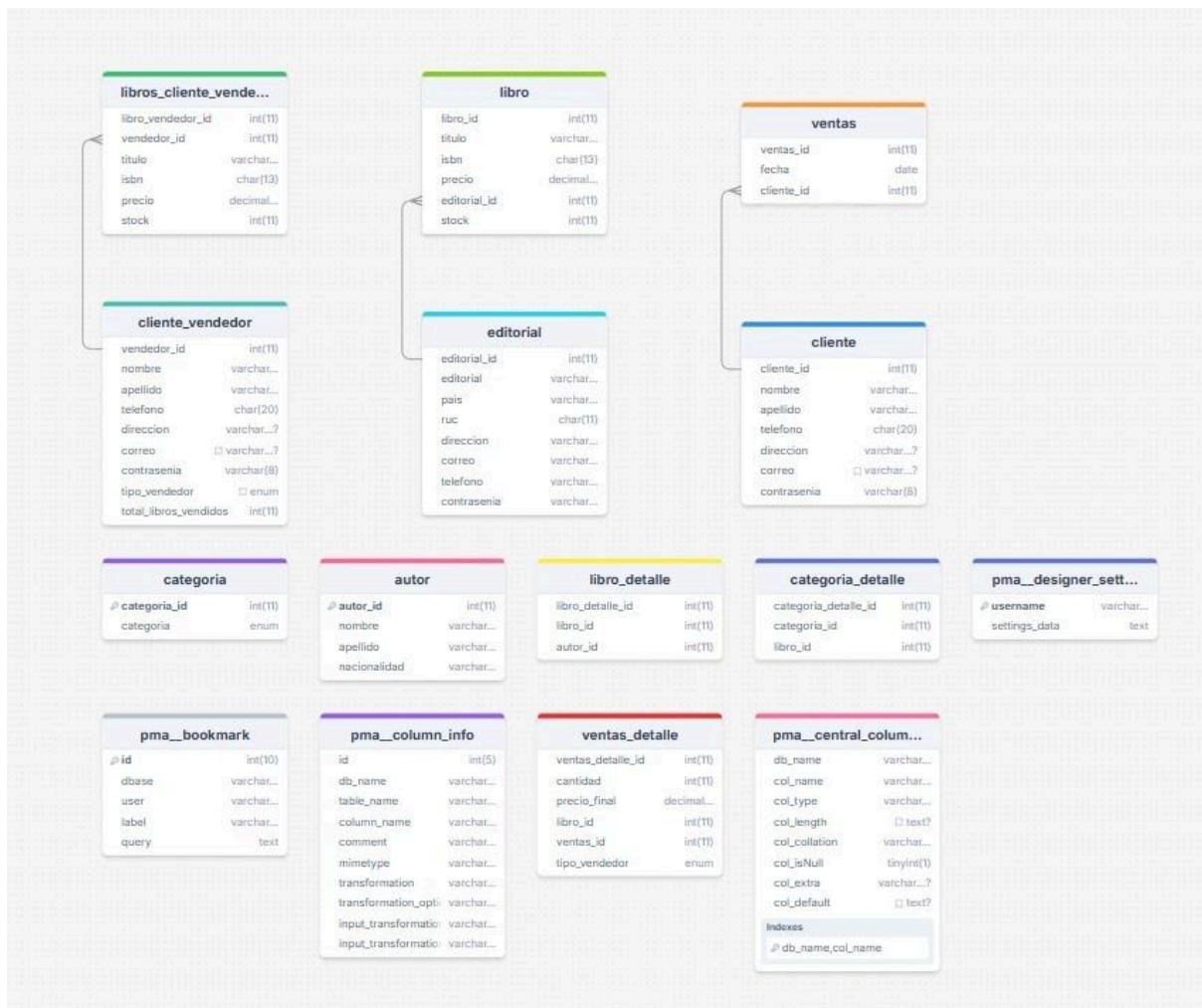
I. Introducción.....	2
II. Base de datos.....	2
A. Modelo Relacional.....	2
III. Consultas CRUD.....	3
A. INSERT.....	3
1. FRONTED.....	3
2. BACKEND.....	6
B. SELECT.....	11
1. FRONTED.....	11
2. BACKEND Consulta SELECT:.....	14
C. UPDATE.....	16
1. FRONTED.....	16
2. BACKEND.....	17
D. DELETE.....	18
1. FRONTED.....	18
2. BACKEND.....	21

# I. Introducción

*El Refugio del Lector* es una librería en línea que combina pasión por los libros con una experiencia interactiva para lectores y escritores. Aquí puedes explorar una amplia variedad de géneros, desde ficción hasta literatura infantil, descubrir las últimas novedades y aprovechar ofertas especiales. Además, si eres un escritor independiente, puedes vender tus libros como vendedor particular o profesional. Con un carrito de compras fácil de usar, se adapta tanto a clientes ocasionales como a amantes de la lectura más fieles, creando una comunidad donde los libros siempre son protagonistas.

## II. Base de datos

### A. Modelo Relacional



### III. Consultas CRUD

#### A. INSERT

Para la consulta INSERT, la hemos decidido poner en el formulario, para que cuando un usuario quiera registrarse se ejecute el INSERT quedé registrado a la base de datos.

#### 1. FRONTED

The screenshot shows two pages of the El Refugio del Lector website. The top page is a registration form titled "Crea tu cuenta" (Create your account). It includes fields for Nombre (Name) containing "Roberth", Apellidos (Last Name) containing "Horna", Teléfono (Phone) containing "667996854", Dirección (Address) containing "direccion", Email containing "netacc138@ijstreaming.com", Contraseña (Password) containing ".....", and Confirmar Contraseña (Confirm Password) containing ".....". There is a checked checkbox for "He leído y acepto la Política de Privacidad" (I have read and accept the Privacy Policy) and a red "Registrarme" (Register) button. The bottom page shows a blurred background of bookshelves with the text "Bienvenido Roberth" (Welcome Roberth) overlaid. Below it is a "Conocer más" (Know more) button. At the bottom, there are four book covers: "DON QUIJOTE DE LA MANCHA" by Miguel de Cervantes, "The GREAT GATSBY", "MARY", and "The Eye".

Código:



```
1  <?php
2  session_start();
3
4 // Determinar el tipo de registro
5 $tipoRegistro = '';
6 if (isset($_POST['registro'])) {
7     $tipoRegistro = "cliente";
8 } else {
9     $tipoRegistro = "vendedor";
10 }
11
12 ?>
13 <!DOCTYPE html>
14 <html lang="en">
15 <head>
16     <meta charset="UTF-8">
17     <meta name="viewport" content="width=device-width, initial-scale=1.0">
18     <title>Formulario de Registro</title>
19     <link rel="stylesheet" href="../css/css_registrarme.css">
20     <link rel="stylesheet" href="../css/cart.css">
21 </head>
22 <body>
23 <header>
24     <nav>
25         <div class="logo">
26             <a href="#">El Refugio del Lector</a>
27         </div>
28         <div class="search-bar">
29             <input type="text" placeholder="Buscar libros...">
30             <button type="button">🔍</button>
31         </div>
32         <div class="nav-links">
33             <ul>
34                 <li><a href="../index.php">Inicio</a></li>
35                 <li><a href="libros.php">Libros</a></li>
36                 <li><a href="#">eBooks</a></li>
37                 <li><a href="#">Papelería</a></li>
38                 <li><a href="#">Novedades</a></li>
39                 <li><a href="#">Ofertas</a></li>
40             </ul>
41         </div>
42         <div class="user-icons">
43             <button type="button" id="view-cart">🛒</button>
44             <div class="dropdown">
45                 <button type="button" class="dropdown-button">👤</button>
46                 <div class="dropdown-content">
47                     <?php if (!empty($tipo_usuario)): ?>
48                     <?php if ($tipo_usuario === 'cliente'): ?>
49                         <a href="#">Mis Pedidos</a>
50                         <a href="cerrar_sesion.php">Cerrar Sesión</a>
51                     <?php else: ?>
52                         <a href="#">Mis Pedidos</a>
53                         <a href="area_vendedor.php">Área de Vendedor</a>
54                         <a href="cerrar_sesion.php">Cerrar Sesión</a>
55                     <?php endif;?>
56                     <?php else:>
57                         <a href="inicio_sesion.php">Iniciar Sesión</a>
58                         <a href="tipo_registro.php">Registrarme</a>
59                         <a href="inicio_sesion.php">Mis Pedidos</a>
60                         <a href="inicio_sesion.php">Área de Vendedor</a>
61                     <?php endif;?>
62                 </div>
63             </div>
64         </div>
65     </nav>
```

```

1   </header>
2   <section>
3
4   <form action="verificacion_registro.php" method="POST">
5
6       <h1>Crea tu cuenta</h1>
7
8       <label for="">Nombre</label>
9       <input type="text" name="name">
10
11      <label for="">Apellidos</label>
12      <input type="text" name="lastName">
13
14      <label for="">Teléfono</label>
15      <input type="text" name="telefono">
16
17      <label for="">Dirección</label>
18      <input type="text" name="direccion">
19
20      <label for="">Email</label>
21      <input type="email" name="email" placeholder="example@hotmail.com">
22
23      <label for="">Contraseña</label>
24      <input type="password" name="password">
25
26      <label for="">Confirmar Contraseña</label>
27      <input type="password" name="verify">
28
29      <label for="">
30          <input type="checkbox" name="policies">
31          He leído y acepto la Política de Privacidad
32      </label>
33
34      <!-- Campo oculto para determinar el tipo de registro -->
35      <input type="hidden" name="tipoRegistro" value="php htmlspecialchars($tipoRegistro) ?&gt;"&gt;
36
37      &lt;input type="submit" value="Registrarme"&gt;
38  &lt;/form&gt;
39  &lt;/section&gt;
40  &lt;footer&gt;
41      &lt;div class="footer-content"&gt;
42          &lt;p&gt;&amp;copy; 2024 El Refugio del Lector. Todos los derechos reservados.&lt;/p&gt;
43          &lt;div class="social-icons"&gt;
44              &lt;a href="#"&gt;Facebook&lt;/a&gt;
45              &lt;a href="#"&gt;Twitter&lt;/a&gt;
46              &lt;a href="#"&gt;Instagram&lt;/a&gt;
47          &lt;/div&gt;
48      &lt;/div&gt;
49  &lt;/footer&gt;
50
51  &lt;!-- Modal del carrito --&gt;
52  &lt;div class="modal" id="cartModal"&gt;
53      &lt;div class="modal-content"&gt;
54          &lt;h2&gt;Carrito de Compras&lt;/h2&gt;
55          &lt;ul class="cart-items" id="cartItems"&gt;&lt;/ul&gt;
56          &lt;button class="delete-cart" id="clearCart"&gt;Vaciar carrito&lt;/button&gt;
57          &lt;button class="close-btn" id="closeModal"&gt;Cerrar&lt;/button&gt;
58      &lt;/div&gt;
59  &lt;/div&gt;
60  &lt;script src="../js/cart.js"&gt;&lt;/script&gt;
61  &lt;/body&gt;
62  &lt;/html&gt;
63
64 &lt;/body&gt;
65 &lt;/html&gt;
</pre

```

## 2. BACKEND

### Consulta INSERT:

```
● ● ●  
1 // Verificar si el cliente ya existe por correo  
2 $sql = "SELECT * FROM cliente WHERE correo = :email";  
3 $stmt = $miPDO->prepare($sql);  
4 $stmt->bindParam(':email', $email);  
5 $stmt->execute();  
6
```

```

1 <?php
2
3 include_once 'conexion.php';
4 ini_set('display_errors', 1);
5 ini_set('display_startup_errors', 1);
6 error_reporting(E_ALL);
7
8
9 if ($_SERVER['REQUEST_METHOD'] == 'POST' && !empty($_POST)) {
10
11     // Recogida de la información del formulario
12     $errores = [];
13     $name = $_POST['name'] ?? '';
14     $lastName = $_POST['lastName'] ?? NULL;
15     $telephone = $_POST['telefono'] ?? NULL;
16     $address = $_POST['direccion'] ?? NULL;
17     $email = $_POST['email'] ?? NULL;
18     $password = $_POST['password'] ?? NULL;
19     $passwordVerify = $_POST['verify'] ?? NULL;
20     $policies = isset($_POST['policies']);
21     $recordType = $_POST['tipoRegistro'] ?? NULL;
22     // elimina espacios en blanco al inicio y al final
23     $email = trim($email);
24     // convierte un string a minúsculas
25     $email = strtolower($email);
26     $tipoRegistro=$_POST['tipoRegistro'];
27 }
28
29 //-----
30 // FUNCIONES DE VALIDACIÓN
31 //-----
32
33 function validateText(string $text): bool
34 {
35     $text = trim($text);
36     return !empty($text) && preg_match('/^([a-zA-ZáéíóúÁÉÍÓÚññ\s]+)$/', $text) && strlen($text) <= 25;
37 }
38
39 function validarTelefono(string $number): bool
40 {
41     $number = preg_replace('/[\s\-\_]/', '', $number);
42     return preg_match('/^\d{9,13}$/', $number) && preg_match('/^(6|7|8|9)\d{8}|\^(\+34)?(6|7|8|9)\d{8}$/', $number);
43 }
44
45 function validateEmail(string $text): bool
46 {
47     return filter_var($text, FILTER_VALIDATE_EMAIL);
48 }
49
50 function validatePassword(string $text): bool
51 {
52     return !empty($text) &&
53         strlen($text) >= 4 &&
54         strlen($text) <= 8 &&
55         preg_match('/[a-z]/', $text) &&
56         preg_match('/[A-Z]/', $text) &&
57         preg_match('/\d/', $text) &&
58         preg_match('/[\W_]/', $text);
59 }
60
61 function validateVerification(string $text1, string $text2): bool
62 {
63     return !empty($text1) && !empty($text2) && $text1 === $text2;
64 }

```

```
1 // VALIDACIONES
2 //-----
3
4 if (!validateText($name)) {
5     $errores[] = "El campo nombre es obligatorio, no puede estar vacío y solo puede contener letras.";
6 }
7
8 if (!validateText($lastName)) {
9     $errores[] = "El campo apellidos es obligatorio, no puede estar vacío y solo puede contener letras.";
10 }
11
12
13 if (!validarTelefono($telephone)) {
14     $errores[] = "El teléfono debe contener 9 a 13 dígitos y tener un formato válido.";
15 }
16
17 if (!validateEmail($email)) {
18     $errores[] = "El email no cuenta con el formato correcto, por favor siga el ejemplo.";
19 }
20
21
22 if (!validatePassword($password)) {
23     $errores[] = "La contraseña debe tener entre 4 y 8 caracteres. Debe incluir al menos un número, una letra minúscula, una letra mayúscula y un carácter especial.";
24 }
25
26 if (!validateVerification($password, $passwordVerify)) {
27     $errores[] = "Las contraseñas no coinciden, por favor inténtelo nuevamente.";
28 }
29
30 if (!$policies) {
31     $errores[] = "Por favor primero acepte las políticas de privacidad.";
32 }
33
34 if (empty($errores)) {
35     if ($tipoRegistro === 'cliente') {
36         // Verificar si el cliente ya existe por correo
37         $sql = "SELECT * FROM cliente WHERE correo = :email";
38         $stmt = $miPDO->prepare($sql);
39         $stmt->bindParam(':email', $email);
40         $stmt->execute();
41
42         if ($stmt->rowCount() > 0) {
43             echo "El cliente con este correo ya existe.";
44             header("Refresh: 3, tipo_registro.php");
45             exit; // Finalizar el script
46         } else {
47             // Insertar nuevo cliente
48             $sql = "INSERT INTO cliente (nombre, apellido, telefono, direccion, correo, contrasenia)
49                   VALUES (:nombre, :apellido, :telefono, :direccion, :correo, :contrasenia)";
50             $stmt = $miPDO->prepare($sql);
51             $stmt->bindParam(':nombre', $name);
52             $stmt->bindParam(':apellido', $lastName);
53             $stmt->bindParam(':telefono', $telephone);
54             $stmt->bindParam(':direccion', $address);
55             $stmt->bindParam(':correo', $email);
56             $stmt->bindParam(':contrasenia', $password);
```



```
1      try {
2          $stmt->execute();
3          echo "<p>Cliente agregado exitosamente!</p>";
4          // Iniciar sesión para el cliente
5          session_name("sesion_de_usuario");
6          session_start();
7          $_SESSION['nombre'] = $name;
8          $_SESSION['tipo_usuario'] = 'cliente';
9          header("Location: ../index.php");
10         exit;
11
12         exit; // Finalizar el script
13     } catch (PDOException $e) {
14         echo "<p>Error al insertar cliente: </p>" . $e->getMessage();
15         exit; // Finalizar el script
16     }
17 }
18 } elseif ($tipoRegistro === 'vendedor') {
19     // Verificar si el cliente existe por correo
20     $sql = "SELECT cliente_id FROM cliente WHERE correo = :email";
21     $stmt = $miPDO->prepare($sql);
22     $stmt->bindParam(':email', $email);
23     $stmt->execute();
24
25     $cliente = $stmt->fetch(PDO::FETCH_ASSOC);
26
27     if ($cliente) {
28         $cliente_id = $cliente['cliente_id'];
29
30         // Verificar si ya está registrado como vendedor
31         $sql = "SELECT * FROM cliente_vendedor WHERE cliente_id = :cliente_id";
32         $stmt = $miPDO->prepare($sql);
33         $stmt->bindParam(':cliente_id', $cliente_id);
34         $stmt->execute();
35
36         if ($stmt->rowCount() > 0) {
37             echo "Este cliente ya está registrado como vendedor.";
38             header("Refresh: 3, tipo_registro.php");
39             exit; // Finalizar el script
40         } else {
41             // Registrar cliente como vendedor
42             $sql = "INSERT INTO cliente_vendedor (cliente_id, tipo_vendedor)
43                         VALUES (:cliente_id, 'particular')";
44             $stmt = $miPDO->prepare($sql);
45             $stmt->bindParam(':cliente_id', $cliente_id);
46
47             try {
48                 $stmt->execute();
49                 echo "<p>Vendedor agregado exitosamente!</p>";
50                 exit; // Finalizar el script
51             } catch (PDOException $e) {
52                 echo "<p>Error al insertar vendedor: </p>" . $e->getMessage();
53                 exit; // Finalizar el script
54             }
55         }
56     } else {
57         echo "El cliente no existe. Regístraloo primero como cliente.";
58         exit; // Finalizar el script
59     }
60 } else {
61     die("Tipo de registro no válido.");
62 }
63 }
```

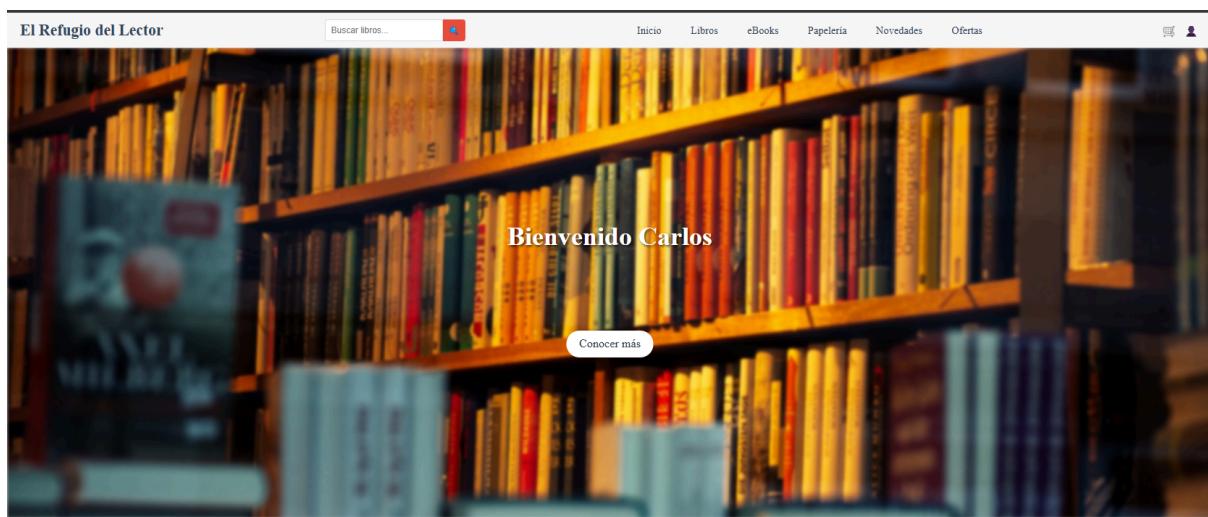
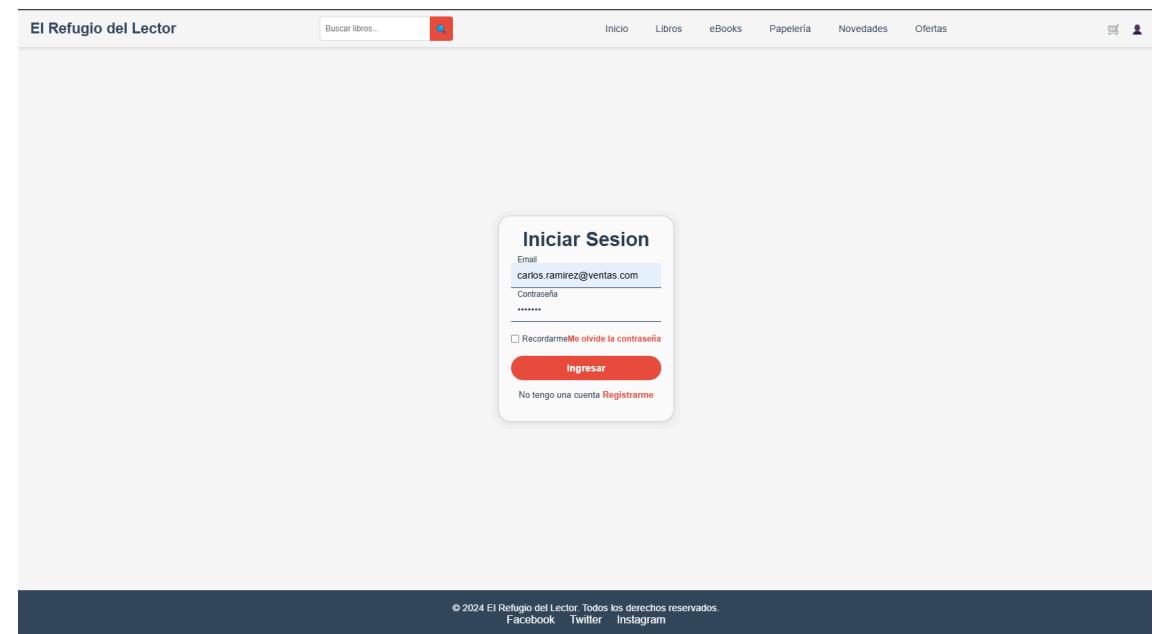


```
1  ?>
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5  <meta charset="UTF-8">
6  <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  <link rel="stylesheet" href="../css/css_errores.css">
8  <title>Document</title>
9  </head>
10 <body>
11
12 <?php
13 if(!empty($errores)){
14
15     echo "<div class='error-messages'>";
16     echo "<ul>";
17
18     foreach($errores as $error):
19         echo "<li> $error </li>";
20     endforeach;
21     header("Refresh:4,tipo_registro.php");
22     echo "</ul>";
23     echo "</div>";
24
25
26 }
27
28
29 ?>
30
31 </body>
32 </html>
```

## B. SELECT

Para la consulta de SELECT nuestro grupo ha optado por hacer la consulta en el inicio de sesión de la página web, en este se usa la consulta anteriormente mencionada para verificar si el correo del usuario y contraseña existe en la base de datos.

### 1. FRONTED



Código:

```
1 <?php
2 session_name("sesion_de_usuario");
3 session_start();
4 $tipo_usuario = isset($_SESSION['tipo_usuario']) ? $_SESSION['tipo_usuario'] : '';
5 $nombre_usuario = isset($_SESSION['nombre']) ? $_SESSION['nombre'] : '';// Aseguramos que exista 'nombre'
6 ?>
7 <!DOCTYPE html>
8 <html lang="en">
9 <head>
10    <meta charset="UTF-8">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12    <title>Iniciar Sesión</title>
13    <link rel="stylesheet" href="../css/css_iniciar_sesion.css">
14    <link rel="stylesheet" href="../css/cart.css">
15 </head>
16 <body>
17   <header>
18     <nav>
19       <div class="logo">
20         <a href="../index.php">El Refugio del Lector</a>
21       </div>
22       <div class="search-bar">
23         <input type="text" placeholder="Buscar libros...">
24         <button type="button">🔍</button>
25       </div>
26       <div class="nav-links">
27         <ul>
28           <li><a href="../index.php">Inicio</a></li>
29           <li><a href="libros.php">Libros</a></li>
30           <li><a href="#">eBooks</a></li>
31           <li><a href="#">Papelería</a></li>
32           <li><a href="#">Novedades</a></li>
33           <li><a href="#">Ofertas</a></li>
34         </ul>
35       </div>
```



```
1      <div class="user-icons">
2          <button type="button" id="view-cart">🛒</button>
3          <div class="dropdown">
4              <button type="button" class="dropdown-button">👤</button>
5              <div class="dropdown-content">
6                  <?php if (!empty($tipo_usuario)): ?>
7                      <?php if ($tipo_usuario == 'cliente'): ?>
8                          <a href="#">Mis Pedidos</a>
9                          <a href="cerrar_sesion.php">Cerrar Sesión</a>
10                     <?php else: ?>
11                         <a href="#">Mis Pedidos</a>
12                         <a href="area_vendedor.php">Área de Vendedor</a>
13                         <a href="cerrar_sesion.php">Cerrar Sesión</a>
14                     <?php endif; ?>
15                 <?php else: ?>
16                     <a href="inicio_sesion.php">Iniciar Sesión</a>
17                     <a href="tipo_registro.php">Registrarme</a>
18                     <a href="inicio_sesion.php">Mis Pedidos</a>
19                     <a href="inicio_sesion.php">Área de Vendedor</a>
20                 <?php endif; ?>
21             </div>
22         </div>
23     </div>
24 </nav>
25 </header>
26 <section>
27     <form action="procesar_inicio_sesion.php" method="post">
28         <h1>Iniciar Sesión</h1>
29         <div class="inputbox">
30             <ion-icon name="mail-outline"></ion-icon>
31             <input type="email" name="email">
32             <label for="">Email</label>
33         </div>
34         <div class="inputbox">
35             <ion-icon name="lock-closed-outline"></ion-icon>
36             <input type="password" name="password">
37             <label for="">Contraseña</label>
38         </div>
39         <div class="forget">
40             <label for=""><input type="checkbox" checked="">Recordarme</label>
41             <a href="#">Me olvide la contraseña</a>
42         </div>
43         <button type="submit">Ingresar</button>
44         <div class="register">
45             <p>No tengo una cuenta <a href="#">Registrarme</a></p>
46         </div>
47     </form>
48 </section>
49 <footer>
50     <div class="footer-content">
51         <p>© 2024 El Refugio del Lector. Todos los derechos reservados.</p>
52         <div class="social-icons">
53             <a href="#">Facebook</a>
54             <a href="#">Twitter</a>
55             <a href="#">Instagram</a>
56         </div>
57     </div>
58 </footer>
59
60     <!-- Modal del carrito -->
61     <div class="modal" id="cartModal">
62         <div class="modal-content">
63             <h2>Carrito de Compras</h2>
64             <ul class="cart-items" id="cartItems"></ul>
65             <button class="delete-cart" id="clearCart">Vaciar carrito</button>
66             <button class="close-btn" id="closeModal">Cerrar</button>
67         </div>
68     </div>
69     <script src="../js/cart.js"></script>
70 </body>
71 </html>
```

## 2. BACKEND

### Consulta SELECT:

```
1 $stmt = $miPDO->prepare('SELECT * FROM cliente WHERE correo = :correo');
2         $stmt->bindParam(':correo', $email, PDO::PARAM_STR);
3         $stmt->execute();
4         $usuario = $stmt->fetch(PDO::FETCH_ASSOC); // Obtener el usuario
5 
```

```
1 <?php
2
3 // Guardar sesión
4 session_name("sesion_de_usuario");
5 session_start();
6
7 require 'conexion.php';
8
9 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
10
11     // Verificar si los datos son nulos o no
12     $email = isset($_POST['email']) ? trim($_POST['email']) : null;
13     $password = isset($_POST['password']) ? trim($_POST['password']) : null;
14
15     if ($email && $password) {
16         try {
17             // Primero, intentamos encontrar al usuario en la tabla 'cliente'
18             $stmt = $miPDO->prepare('SELECT * FROM cliente WHERE correo = :correo');
19             $stmt->bindParam(':correo', $email, PDO::PARAM_STR);
20             $stmt->execute();
21             $usuario = $stmt->fetch(PDO::FETCH_ASSOC); // Obtener el usuario
22
23             if ($usuario) {
24                 // Verificar si la contraseña es correcta
25                 if ($usuario['contrasenia'] === $password) {
26                     // Iniciar sesión de manera segura
27                     session_regenerate_id(true); // Previene fijación de sesión
28                     $_SESSION['email'] = $usuario['correo'];
29                     $_SESSION['nombre'] = $usuario['nombre'];
30
31                     // Establecer que el usuario es un cliente
32                     $_SESSION['tipo_usuario'] = 'cliente';
33
34                     // Redirigir al index
35                     header('location: ../index.php');
36                     exit;
37                 } else {
38                     // Contraseña incorrecta
39                     echo "<p><strong>Contraseña incorrecta.</strong></p>";
40                 }
41             }
42         }
43     }
44 }
```

```

1      } else {
2          // Si no es un cliente, verificar si es un cliente vendedor
3          $stmt = $miPDO->prepare('SELECT * FROM cliente_vendedor WHERE correo = :correo');
4          $stmt->bindParam(':correo', $email, PDO::PARAM_STR);
5          $stmt->execute();
6          $vendedor = $stmt->fetch(PDO::FETCH_ASSOC); // Obtener el cliente vendedor
7
8          if ($vendedor) {
9              // Verificar si la contraseña es correcta
10             if ($vendedor['contraseña'] === $password) {
11                 // Iniciar sesión de manera segura
12                 session_regenerate_id(true); // Previene fijación de sesión
13                 $_SESSION['email'] = $vendedor['correo'];
14                 $_SESSION['nombre'] = $vendedor['nombre'];
15                 // Establecer que el usuario es un cliente vendedor
16                 $_SESSION['tipo_usuario'] = 'vendedor';
17                 //-----Recuperar ID-----
18
19                 $stmt = $miPDO->prepare("SELECT vendedor_id FROM cliente_vendedor WHERE correo = :correo");
20                 $stmt->bindParam(':correo', $email, PDO::PARAM_STR);
21                 $stmt->execute();
22                 $vendedor_id = $stmt->fetchColumn();
23
24                 if ($vendedor_id) {
25                     $_SESSION['id_vendedor'] = $vendedor_id;
26                     echo "El ID del vendedor es: " . $vendedor_id;
27                 } else {
28                     echo "No se encontró un vendedor con ese correo.";
29                 }
30
31
32             //-----
33
34
35             // Redirigir al index
36             header('location: ../index.php');
37             exit;
38         } else {
39             // Contraseña incorrecta
40             echo "<p><strong>Contraseña incorrecta.</strong></p>";
41         }
42     } else {
43         // Usuario no encontrado en ninguna de las tablas
44         echo "<p><strong>El usuario no existe.</strong></p>";
45     }
46 }
47
48 } catch (PDOException $e) {
49     // Manejo seguro de errores
50     error_log("Error de conexión: " . $e->getMessage());
51     echo "<p>Ocurrió un error. Por favor, inténtalo más tarde.</p>";
52 }
53 } else {
54     echo "<p><strong>Por favor, completa todos los campos.</strong></p>";
55 }
56 } else {
57     // Si no se recibieron datos del formulario
58     echo "<p>ERROR: No se recibieron datos del formulario.</p>";
59 }
60
61 ?>
62

```

## C. UPDATE

La actualización de datos se ha aplicado en el área de vendedor , la sentencia nos permite actualizar todos los datos o algunos de ellos de los libros que han sido añadidos por el vendedor

### 1. FRONTED

	libro_vendedor_id	vendedor_id	titulo	isbn	precio	stock
<a href="#">Actualizar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	1	1	El Principito	9782070612758	10.99	10

### Actualizar Libro

1
Alicia en el país de las maravillas
20.00
15
<a href="#">actualizar</a>

Base de datos, tablas y trigger creados correctamente. Datos insertados correctamente. Los datos del libro se han actualizado correctamente.

libro_vendedor_id	vendedor_id	titulo	isbn	precio	stock
1	1	Alicia en el país de las maravillas	9782070612758	20.00	15

## 2. BACKEND

```
● ● ●
1 case 'actualizar':
2
3     $libro_id = $_POST['id_Libro'] ?? NULL; // Corregido el espacio extra
4     $nuevoTitulo = $_POST['titulo'] ?? NULL;
5     $nuevoPrecio = $_POST['precio'] ?? NULL;
6     $nuevoStock = $_POST['stock'] ?? NULL;
7
8
9     // Verificar que el ID del libro es obligatorio
10
11
12     // Construir partes dinámicas de la consulta
13     $updates = [];
14     $params = [];
15
16     if (!empty($nuevoTitulo)) {
17         $updates[] = "titulo = :titulo";
18         $params[':titulo'] = $nuevoTitulo;
19     } elseif ($nuevoTitulo === '') { // Verifica si el campo está vacío explicitamente
20         $errores[] = "El título no puede estar vacío.";
21     }
22
23     if (!empty($nuevoPrecio) && $nuevoPrecio > 0) {
24         $updates[] = "precio = :precio";
25         $params[':precio'] = $nuevoPrecio;
26     } elseif (!empty($nuevoPrecio) && $nuevoPrecio <= 0) {
27         $errores[] = "El precio debe ser mayor a 0.";
28     }
29
30     if (!empty($nuevoStock) && $nuevoStock >= 0) { // Incluye el stock igual a 0 como válido
31         $updates[] = "stock = :stock";
32         $params[':stock'] = $nuevoStock;
33     } elseif (!empty($nuevoStock) && $nuevoStock < 0) {
34         $errores[] = "El stock no puede ser negativo.";
35     }
36
37     // Verificar errores
38     if (!empty($errores)) {
39         foreach ($errores as $error) {
40             echo "<p>Error: $error</p>";
41         }
42         exit;
43     }
44
45     // Verificar que hay campos para actualizar
46     if (empty($updates)) {
47         echo "No hay datos para actualizar.";
48         exit;
49     }
50
51
52     // Construir la consulta SQL
53     $query = "UPDATE libros_cliente_vendedor
54         SET " . implode(', ', $updates) . "
55         WHERE libro_vendedor_id = :libro_id";
56
57     $params[':Libro_id'] = $libro_id;
58
59     // Ejecutar la consulta
60     try {
61         $stmt = $miPDO->prepare($query);
62         $stmt->execute($params);
63
64         echo "Los datos del libro se han actualizado correctamente.";
65         header("Refresh: 4; url=area_vendedor.php");
66         exit;
67     } catch (PDOException $e) {
68         echo "Error al actualizar: " . $e->getMessage();
69     }
70     break;
71
```

## D. DELETE

En el caso de la consulta DELETE nuestro grupo le pareció que era mejor opción hacer la consulta para que los usuarios puedan eliminar cualquier libro que ya no quieran vender los usuarios.

### 1. FRONTEDE

**Área de Vendedor**

**Añadir Libro**

**Actualizar Libro**

**Eliminar Libro**

© 2024 El Refugio del Lector. Todos los derechos reservados.  
Facebook Twitter Instagram

Base de datos, tablas y trigger creados correctamente. Datos insertados correctamente. eliminarLibro eliminado correctamente.

Verificamos en la Base de Datos que se haya eliminado correctamente el libro de la table **libros\_cliente\_vendedor**.

Código:

```

1  <?php
2  session_name("sesion_de_usuario");
3  session_start();
4  $tipo_usuario = isset($_SESSION['tipo_usuario']) ? $_SESSION['tipo_usuario'] : '';
5  $nombre_usuario = isset($_SESSION['nombre']) ? $_SESSION['nombre'] : '';// Aseguramos que exista 'nombre'
6  ?>
7  <!DOCTYPE html>
8  <html lang="es">
9  <head>
10   <meta charset="UTF-8">
11   <meta name="viewport" content="width=device-width, initial-scale=1.0">
12   <title>El Refugio del Lector</title>
13   <link rel="stylesheet" href="../css/css_area_vendedor.css">
14   <link rel="stylesheet" href="../css/cart.css">
15  </head>
16  <body>
17  <header>
18    <nav>
19      <div class="logo">
20          <a href="../index.php">El Refugio del Lector</a>
21      </div>
22      <div class="search-bar">
23          <input type="text" placeholder="Buscar libros...">
24          <button type="button">🔍</button>
25      </div>
26      <div class="nav-links">
27          <ul>
28              <li><a href="../index.php">Inicio</a></li>
29              <li><a href="libros.php">Libros</a></li>
30              <li><a href="#">eBooks</a></li>
31              <li><a href="#">Papeleria</a></li>
32              <li><a href="#">Novedades</a></li>
33              <li><a href="#">Ofertas</a></li>
34          </ul>
35      </div>
36      <div class="user-icons">
37          <button type="button" id="view-cart">🛒</button>
38          <div class="dropdown">
39              <button type="button" class="dropdown-button">👤</button>
40              <div class="dropdown-content">
41                  <?php if (!empty($tipo_usuario)): ?>
42                      <?php if ($tipo_usuario == 'cliente'): ?>
43                          <a href="#">Mis Pedidos</a>
44                          <a href="cerrar_sesion.php">Cerrar Sesión</a>
45                      <?php else: ?>
46                          <a href="#">Mis Pedidos</a>
47                          <a href="#">Área de Vendedor</a>
48                          <a href="cerrar_sesion.php">Cerrar Sesión</a>
49                      <?php endif; ?>
50                  <?php else:
51                      header('location: ../index.php');
52                      endif; ?>
53                  </div>
54              </div>
55          </div>
56      </nav>
57  </header>

```

```

1  <section class="containerBottom">
2      <h1>Área de Vendedor</h1>
3      <form action="procesar_area_vendedor.php" method="POST">
4          <!-- Añadir libro -->
5          <h3>Añadir Libro</h3>
6          <input type="hidden" name="accion" value="añadir">
7          <input type="text" name="titulo" placeholder="Título del libro" >
8          <input type="text" name="isbn" placeholder="ISBN">
9          <input type="number" name="precio" placeholder="Precio" step="0.01" >
10         <input type="number" name="stock" placeholder="Stock" >
11         <input type="submit" name="accion" value="añadir">
12
13         <!-- Actualizar libro -->
14         <h3>Actualizar Libro</h3>
15
16         <input type="number" name="id_libro" placeholder="ID del libro">
17         <input type="text" name="titulo" placeholder="Nuevo título" >
18         <input type="number" name="precio" placeholder="Nuevo precio" step="0.01" >
19         <input type="number" name="stock" placeholder="Nuevo stock" >
20         <input type="submit" name="accion" value="actualizar">
21
22         <!-- Eliminar libro -->
23         <h3>Eliminar Libro</h3>
24
25         <input type="number" name="libro_id" placeholder="ID del libro" >
26         <input type="submit" name="accion" value="eliminar">
27
28     </form>
29 </section>
30 <footer>
31     <div class="footer-content">
32         <p>&copy; 2024 El Refugio del Lector. Todos los derechos reservados.</p>
33         <div class="social-icons">
34             <a href="#">Facebook</a>
35             <a href="#">Twitter</a>
36             <a href="#">Instagram</a>
37         </div>
38     </div>
39 </footer>
40         <!-- Modal del carrito -->
41         <div class="modal" id="cartModal">
42             <div class="modal-content">
43                 <h2>Carrito de Compras</h2>
44                 <ul class="cart-items" id="cartItems"></ul>
45                 <button class="delete-cart" id="clearCart">Vaciar carrito</button>
46                 <button class="close-btn" id="closeModal">Cerrar</button>
47             </div>
48         </div>
49         <script src="../js/cart.js"></script>
50 </body>

```

## 2. BACKEND

```
1  <?php
2 session_name("sesion_de_usuario");
3 session_start();
4
5 if (!isset($_SESSION['tipo_usuario']) || $_SESSION['tipo_usuario'] != 'vendedor') {
6     header('Location: ../index.php');
7     exit();
8 }
9
10 //-----Funciones de validaciones -----
11 $errores = [];
12
13
14
15 require_once 'conexion.php';
16
17 if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['accion'])) {
18     $accion = $_POST['accion'];
19     echo "$accion";
20
21     switch ($accion) {
22         case 'añadir':
23
24             break;
25         case 'actualizar':
26
27             $libro_id = $_POST['id_libro'] ?? NULL; // Corregido el espacio extra
28             $nuevoTitulo = $_POST['titulo'] ?? NULL;
29             $nuevoPrecio = $_POST['precio'] ?? NULL;
30             $nuevoStock = $_POST['stock'] ?? NULL;
31
32
33
34         // Verificar que el ID del libro es obligatorio
35
36
37         // Construir partes dinámicas de la consulta
38         $updates = [];
39         $params = [];
40
41         if (!empty($nuevoTitulo)) {
42             $updates[] = "titulo = :titulo";
43             $params[':titulo'] = $nuevoTitulo;
44         } elseif ($nuevoTitulo === '') { // Verifica si el campo está vacío explícitamente
45             $errores[] = "El título no puede estar vacío.";
46         }
47
48         if (!empty($nuevoPrecio) && $nuevoPrecio > 0) {
49             $updates[] = "precio = :precio";
50             $params[':precio'] = $nuevoPrecio;
51         } elseif (!empty($nuevoPrecio) && $nuevoPrecio <= 0) {
52             $errores[] = "El precio debe ser mayor a 0.";
53         }
54
55         if (!empty($nuevoStock) && $nuevoStock >= 0) { // Incluye el stock igual a 0 como válido
56             $updates[] = "stock = :stock";
57             $params[':stock'] = $nuevoStock;
58         } elseif (!empty($nuevoStock) && $nuevoStock < 0) {
59             $errores[] = "El stock no puede ser negativo.";
60         }
61     }
}
```

```

1      // Verificar errores
2      if (!empty($errores)) {
3          foreach ($errores as $error) {
4              echo "<p>Error: $error</p>";
5          }
6          exit;
7      }
8
9      // Verificar que hay campos para actualizar
10     if (empty($updates)) {
11         echo "No hay datos para actualizar.";
12         exit;
13     }
14
15     // Construir la consulta SQL
16     $query = "UPDATE libros_cliente_vendedor
17             SET " . implode(', ', $updates) . "
18             WHERE libro_vendedor_id = :libro_id";
19
20     $params[':libro_id'] = $libro_id;
21
22     // Ejecutar la consulta
23     try {
24         $stmt = $miPDO->prepare($query);
25         $stmt->execute($params);
26
27         echo "Los datos del libro se han actualizado correctamente.";
28         header("Refresh: 4; url=area_vendedor.php");
29         exit;
30     } catch (PDOException $e) {
31         echo "Error al actualizar: " . $e->getMessage();
32     }
33     break;
34
35 case 'eliminar':
36     if (!empty($_POST['libro_id'])) {
37         $libro_id = $_POST['libro_id'];
38
39         try {
40             $sql = "DELETE FROM libros_cliente_vendedor WHERE libro_vendedor_id = :libro_id";
41             $stmt = $miPDO->prepare($sql);
42             $stmt->execute([':libro_id' => $libro_id]);
43             echo "Libro eliminado correctamente.";
44             header("Refresh: 3, area_vendedor.php");
45             exit; // Finalizar el script
46         } catch (PDOException $e) {
47             echo "Error al eliminar el libro: " . $e->getMessage();
48         }
49     } else {
50         echo "El ID del libro es obligatorio--.";
51         echo "$accion";
52     }
53     break;
54
55 default:
56     echo "Acción no válida.";
57 }
58 }
59
60 foreach($errores as $error){
61     echo "$error";
62 }
63 ?>
```

