

“Año del Fortalecimiento de la Soberanía Nacional”

UNIVERSIDAD TECNOLÓGICA DE LOS ANDES



ESCUELA DE
INGENIERÍA DE
SISTEMAS E
INFORMÁTICA



Tema: Trabajo grupal, agregando funcionalidad al software de TODOLIST

CURSO: INGENIERIA DE SOFTWARE II

DOCENTE: EDUARDO CHAVEZ VASQUEZ

Integrantes:

- *YONIL MEJIA QUISPE*
- *HIROSHI JAISEN GONZALES ROMAN*
- *GIOVANI SALLO ZAMALLOA*

CUSCO - PERU
2022

Funciones Agregadas:

Tarjetas:

| | |
|--|---|
| | |
| | <p>Función 1: <u>Botton de recuperar contraseña olvidada</u></p> <p>Desarrollado por: Yonil mejia Quispe</p> |

| | |
|--|--|
| | |
| | <p>Función 2: <u>Eliminar cuenta de usuario</u></p> <p>Desarrollado por: Yonil mejia Quispe</p> |

| | |
|--|--|
| | |
| | <p>Función 3: <u>Botton de Contactanos</u></p> <p>Desarrollado por: Hiroshi Jaysen Gonzales Roman</p> |

| | |
|--|--|
| | |
| | <p>Función 4: <u>Botton de Recordar usuario y contraseña</u></p> <p>Desarrollado por: Hiroshi Jaysen Gonzales Roman</p> |

| | |
|--|--|
| | |
| | <p>Función 5: <u>Botton Borrar todo</u></p> <p>Desarrollado por: Giovani Sallo Zamalloa</p> |

| | |
|--|--|
| | |
| | <p>Función 6: <u>Botton Modo oscuro</u></p> <p>Desarrollado por: Giovani Sallo Zamalloa</p> |

Descripciones de las Funcionalidades:

Función 1: Boton de recuperar contraseña olvidada

Objetivo: Recupera la contraseña de un usuario que olvido su contraseña por medio del correo vinculado a su cuenta. Para ello se utilizó la librería de PHPMAILER

1. Para ello se ingresa pulsando el botón de Olvide mi clave:



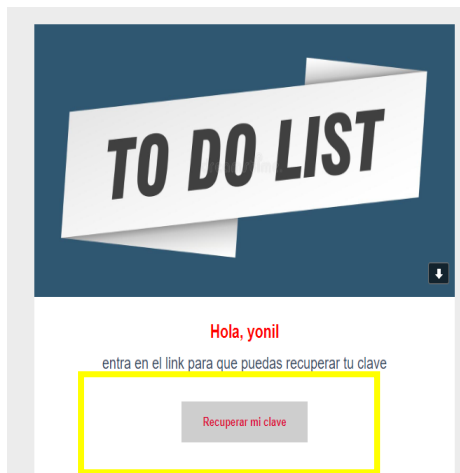
2. Luego se tendra que ingresar el gmail vinculado a la cuenta:



```
<form action="linkClave.php" method="post">
<header> ToDoList</header>
<h1>
| Recuperar tu Clave
</h1>
<div>
| <label>Correo: </label>
| <br>
| <input type="email" name="email" placeholder="Ejemplo@gmail.com" required autocomplete="off" />
</div>
<input type="submit" class="button button-block mi8tn mt-5" value="RECUPERAR CLAVE"/>
<br><br>
<a href="index.php" id="volver" class="mt-3 mb-4" title="Volver">Volver</a>
<br><br>
</form>
</div>
| <h1><?php
session_start();
ob_start();
error_reporting(0);
$msgj=$ _SESSION['msgj'];
unset($_SESSION['msgj']);

if($msgj==1){
| echo "El correo no esta vinculado a ninguna cuenta";
}
```

3. Mediante en el back en se realizará consultas a la base de datos, y se generara un **token de usuario** para la seguridad de usuario, seguidamente se enviará un mensaje Email al correo de usuario con un link para que pueda cambiar la contraseña a una nueva.



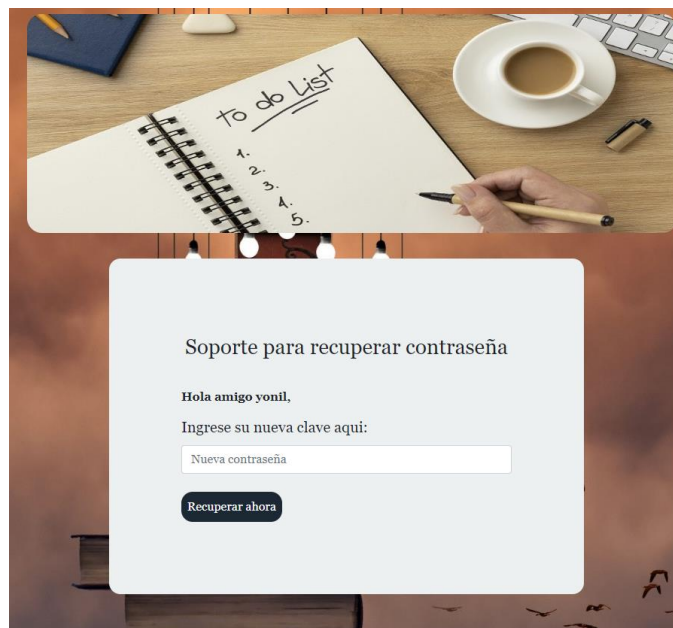
```
try {
    $mail->SMTPOptions = array(
        'ssl' => array(
            'verify_peer' => false,
            'verify_peer_name' => false,
            'allow_self_signed' => true
        )
    );
    //Server settings
    $mail->SMTPDebug = 0; // Enable verbose debug output
    $mail->isSMTP(); // Send using SMTP
    $mail->Host = 'smtp.gmail.com'; // Set the SMTP server to send through
    $mail->SMTPAuth = true; // Enable SMTP authentication
    $mail->Username='201913301c@utea.edu.pe';//este debe ir en el address?
    $mail->Password='*****'; // SMTP password
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS; // Enable TLS encryption; `PHPMailer::`
    $mail->Port = 587; // TCP port to connect to, use 465 for

    //Recipients
    $mail->setFrom('201913301c@utea.edu.pe', 'ToDoList');
    $mail->addAddress($destinatario); // Add a recipient

    // Content
    $mail->isHTML(true); // Set email format to HTML
    $mail->Subject = $asunto;
    $mail->Body = $cuerpo;

    $mail->send();
    $_SESSION['sesion_exito']=6;
    header('location:index.php');
}
```

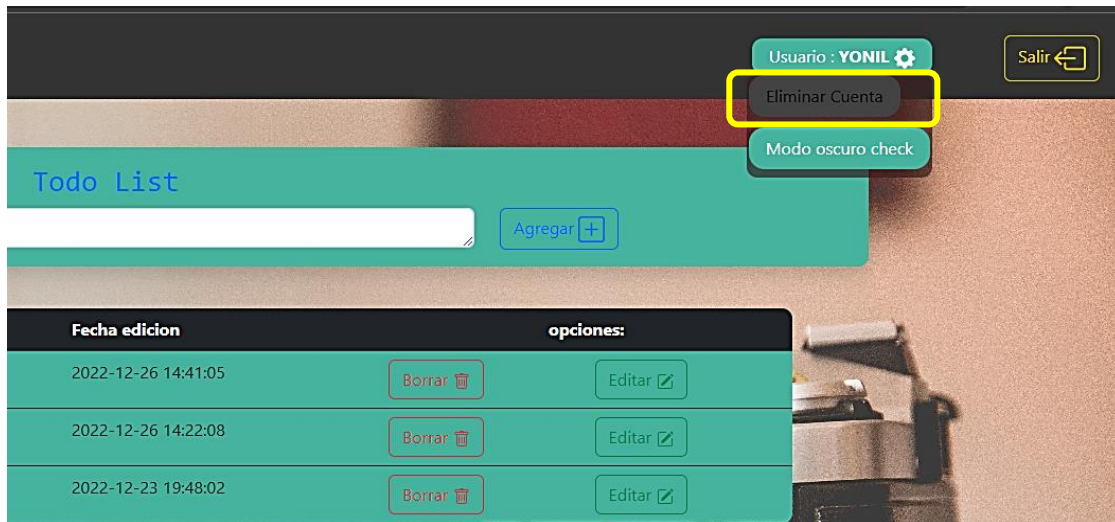
4. Para luego poder **cambiar la contraseña**:



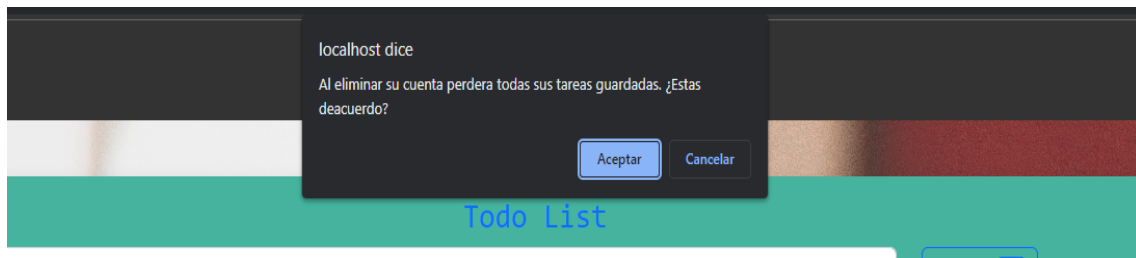
Función 2: **Eliminar cuenta de usuario**

Objetivo: Dar libre elección al usuario de eliminar su cuenta de TodoList con todos los datos en duro para liberar toda relación de datos con el programa del TodoList.

1. Para ello es necesario ingresar con cuenta a la interfaz del Tareas y buscar la **opción de Eliminar cuenta**



2. Seguidamente se enviará una **alerta de advertencia**:



```
183
184 </head>
185 <script type="Text/javascript">
186   function DeleteUser(){
187     var texto= confirm("Al eliminar su cuenta perdera todas sus tareas guardadas. ¿Estas deacuerdo?");
188     if(texto==true){
189       return true;
190     }else{
191       return false;
192     }
193   }
194 </script>
195
196
197
```

3. Donde al **confirmar se elimina todos los datos** relacionados al usuario y las tareas guardas de la base de datos del phpAdmin.

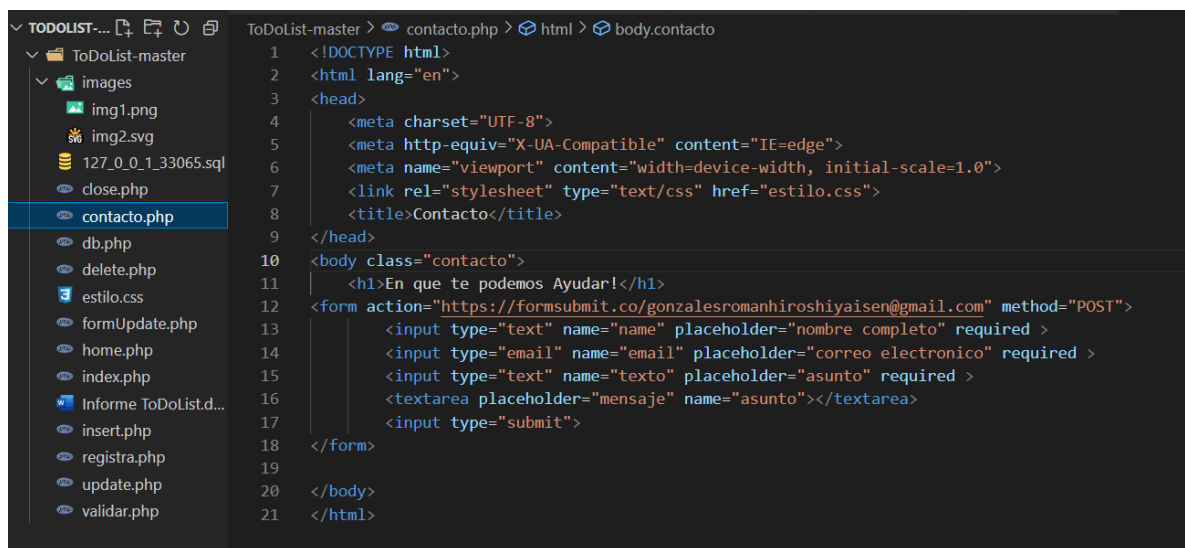
Función 3: Boton de Contactanos

1. Se crea el código dentro del archivo index.php línea 168 y en home.php línea 240 el cual los conectaremos con el archivo contacto.php.

```
168 <a style="text-decoration:none ;" class="contacto" href="contacto.php"> Contactanos</a>  
169
```

```
240 <a style="text-decoration:none ;" class="contacto" href="contacto.php"> Contactanos</a>
```

2. Se crea el archivo contacto.php donde se encuentra la acción de formsuit que nos ayudara que los mensajes enviados lleguen a nuestro correo y también está el formulario para enviar el mensaje en el cual se tiene nombre, correo electrónico, asunto y el mensaje este está conectado a estilo.css para poder darle fondo de pantalla y color a las celdas de la tabla.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the following structure:

- ToDoList-master
 - images
 - img1.png
 - img2.svg
 - 127_0_0_1_33065.sql
 - close.php
 - contacto.php (selected)
 - db.php
 - delete.php
 - estilo.css
 - formUpdate.php
 - home.php
 - index.php
 - Informe ToDoList.d...
 - insert.php
 - registra.php
 - update.php
 - validar.php

The code editor shows the content of 'contacto.php' with line numbers 1 through 21:

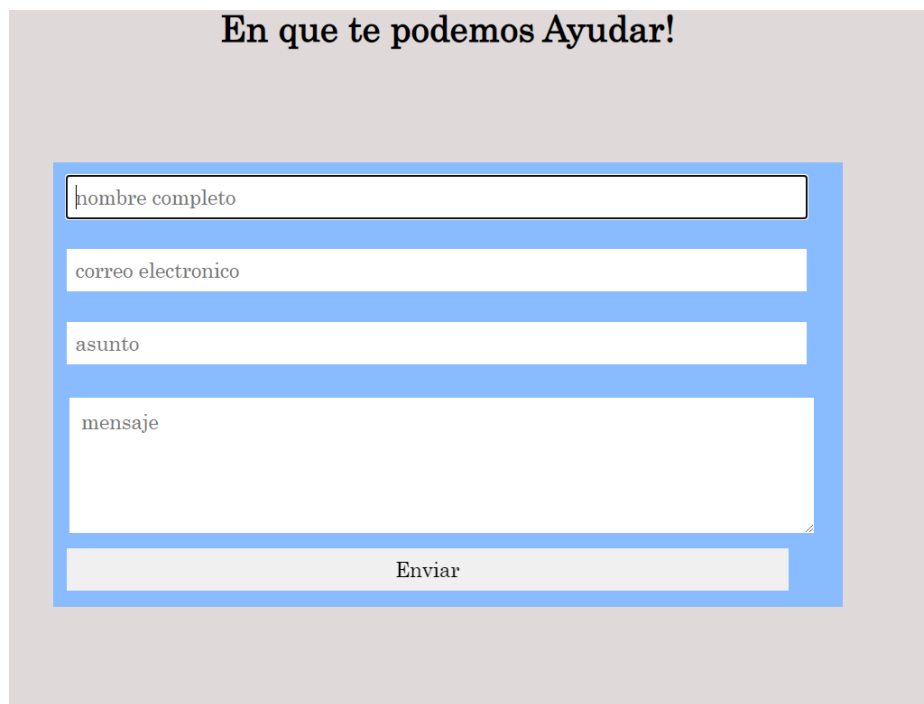
```
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4 <meta charset="UTF-8">  
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7 <link rel="stylesheet" type="text/css" href="estilo.css">  
8 <title>Contacto</title>  
9 </head>  
10 <body class="contacto">  
11 <h1>En que te podemos Ayudar!</h1>  
12 <form action="https://formsuubmit.co/gonzalesromanhirosihiyaisen@gmail.com" method="POST">  
13 <input type="text" name="name" placeholder="nombre completo" required >  
14 <input type="email" name="email" placeholder="correo electronico" required >  
15 <input type="text" name="texto" placeholder="asunto" required >  
16 <textarea placeholder="mensaje" name="asunto"></textarea>  
17 <input type="submit">  
18 </form>  
19  
20 </body>  
21 </html>
```

3. Lo colocamos en la parte inferior de la tabla con el nombre de Contáctanos



The image shows a login interface for 'ToDoList Sistema de Ingreso'. It features a black header with the title 'ToDoList' in white. Below it, the text 'Sistema de Ingreso' is displayed. There are two input fields: 'Usuario:' with the value 'pma' and 'Contraseña:' with masked characters '.....'. A checkbox labeled 'Recordar usuario y contraseña' is present. Below these are three buttons: 'ingresar' (blue), '¡No tengo Cuental' (blue), and 'Olvide mi clave...' (blue). At the bottom, a button labeled 'Contáctanos' is highlighted with a yellow border.

Nos redirigirá a una ventana donde tendremos que **llenar un formulario y enviarlo**



The image shows a contact form titled 'En que te podemos Ayudar!'. It has a light gray background. The form is enclosed in a blue border and contains four input fields: 'nombre completo', 'correo electronico', 'asunto', and 'mensaje'. Below the fields is a button labeled 'Enviar'.

Después de enviarlo nos aparecerá una casilla para marcar que nos ayudara a evitar el spam.



Una vez marcado se enviara el mensaje al correo.

Función 4: Boton de Recordar usuario y contraseña

1. Creamos el código en index.php en el cual crearemos el botón de Recordar usuario y contraseña, luego le agregamos en las líneas de código de el nombre y la contraseña para que los datos ingresados sean enviados a el archivo validar.php

```
index.php      152 <p>Usuario: <input type="text" placeholder="Ingrese usuario" name="usuario" value="<?php echo $id ?>"> </p>
               153 <br>
Informe ToDol... 154 <p>Contraseña: <input type="password" placeholder="ingrese su contraseña" name="password" value="<?php echo $pass ?>"> </p>
insert.php      155
registra.php     156
update.php      157 <label>
               158 | <input type="checkbox" name="recordarme">Recordar usuario y contraseña
validar.php     159 </label>
```

2. Dentro del archivo validar.php Creamos la función isset junto al post con el cual se recibiremos las variables enviadas de el usuario y la contraseña esto solo si se ingresaron los datos de un usuario ya logeado por lo contrario si al momento de entrar sin pulsar el botón Recordar usuario y contraseña los datos se eliminaran.

```
registra.php    32
update.php      33
validar.php     34 if(isset($_POST['recordarme'])){
               35     .....setcookie('usuario',$_POST['usuario'],time() + 60*60*24);
               36     .....setcookie('password',$_POST['password'],time() + 60*60*24);
               37     .....}
               38     .....else{
               39     .....setcookie('usuario','',time() - 60*60*24);
               40     .....setcookie('password','',time() - 60*60*24);
               41     .....}
```

3. Creamos el código en el archivo index.php para que si se pulso en el botón de Recordar nombre de Usuario y Contraseña y una vez salga de su cuenta aparezca con el nombre y usuario que se ingresó anteriormente pero si no lo pulso que elimine el nombre y contraseña.


```

121         if(isset($_COOKIE['usuario']) && isset($_COOKIE['password'])){
122             $id=$_COOKIE['usuario'];
123             $pass=$_COOKIE['password'];
124         }
125         else{
126             $id="";
127             $pass="";
128         }

```

4.Colocaremos el botón en la parte inferior debajo de Contraseña

ToDoList

Sistema de Ingreso

Usuario:

pma

Contraseña:

☐ Recordar usuario y contraseña

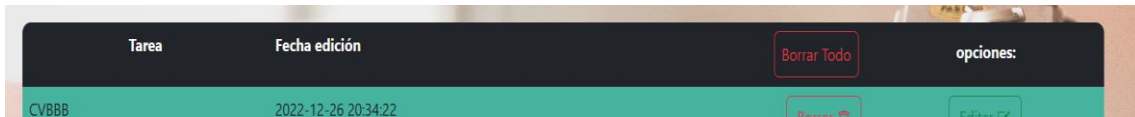
ingresar

¡No tengo Cuenta! Olvide mi clave...

Contactanos

Función 5: Boton de borrar todo

1. Se crea el código dentro del archivo home.php creamos el botón especificando la variable que nos permitirá enviar a nuestro archivo que tendrá la acción correspondiente, igualmente le encontramos un lugar adecuado e indicado.

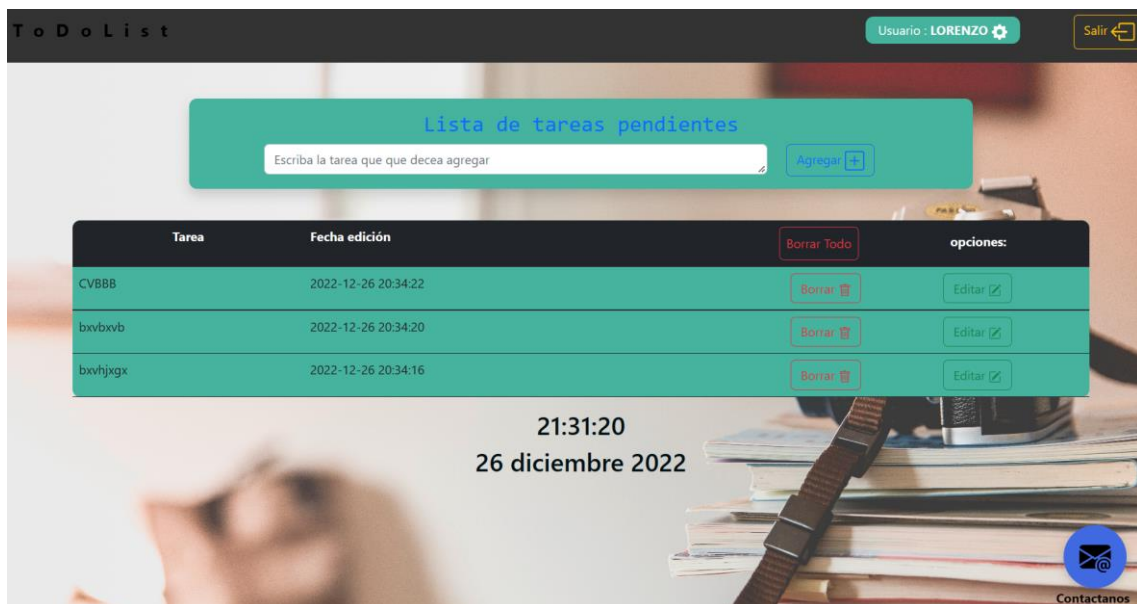


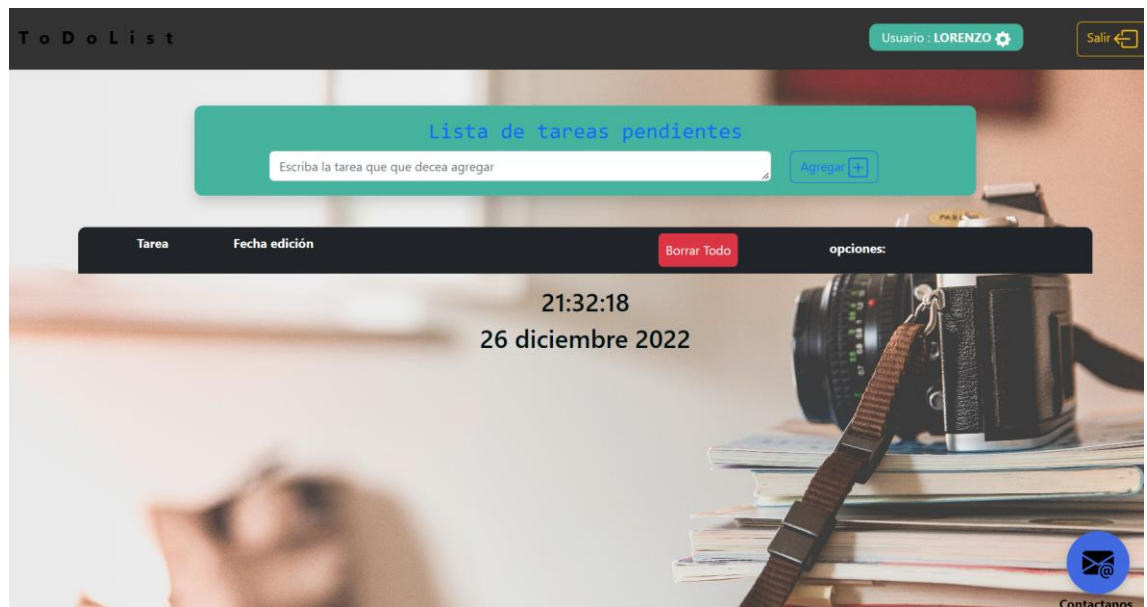
2. Se crea el archivo remove.php donde se encuentra la acción de borrar tabla, conectamos el botón y la variable con la acción que va a realizar que será borrar todas las tareas.

```
remove.php > ...
1 <?php
2 include 'db.php';
3 session_start();
4 | ob_start();
5 $idUser=$_SESSION['id'];
6 mysqli_query($conexion,"DELETE FROM tarea where idUsuario='$idUser'");
7 header("location:home.php");
8
9 ?>
```

Y como vemos funciona correctamente:

1





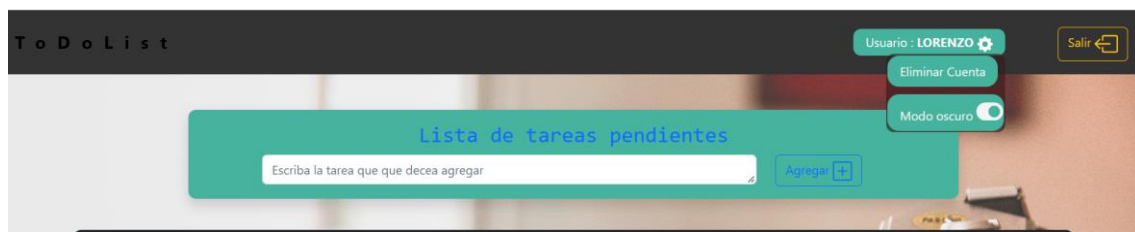
Función 6: Boton de Modo Oscuro

1. Para esta funcionalidad se vio el problema de la vista así que se decidió implementar este modo oscuro, primeramente, ubicamos la posición de la acción que en este caso decidimos ponerlo dentro de usuario opciones.

```

152 <p>Usuario: <input type="text"placeholder="Ingrese usuario" name="usuario" value="<?php echo $id ?>" </p>
153 <br>
154 <p>Contraseña: <input type="password"placeholder="ingrese su contraseña" name="password" value="<?php echo $pass ?>" </p>
155
156
157 <label>
158 | <input type="checkbox" name="recordarme">Recordar usuario y contraseña
159 </label>

```

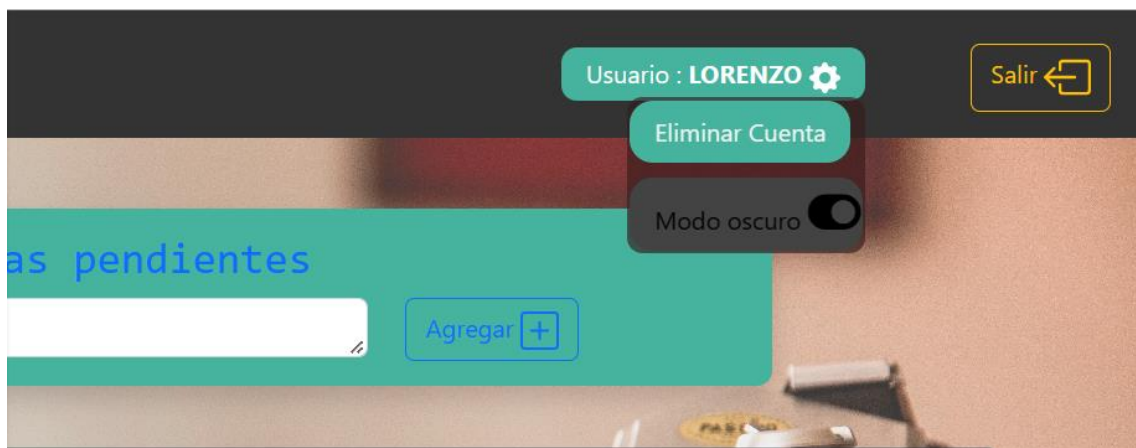
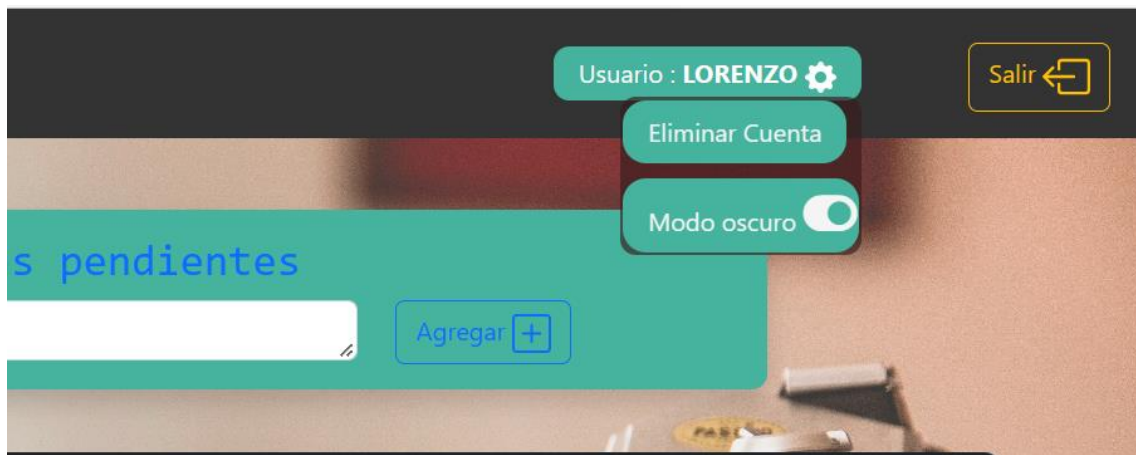


En este caso es un checkbox personalizado.

2. Creamos una carpeta de js y dentro de el un archivo main.js esto para darle funcionalidad al checkbox de prendido y apagado, seguidamente codificamos lo necesario para que este funcione correctamente.

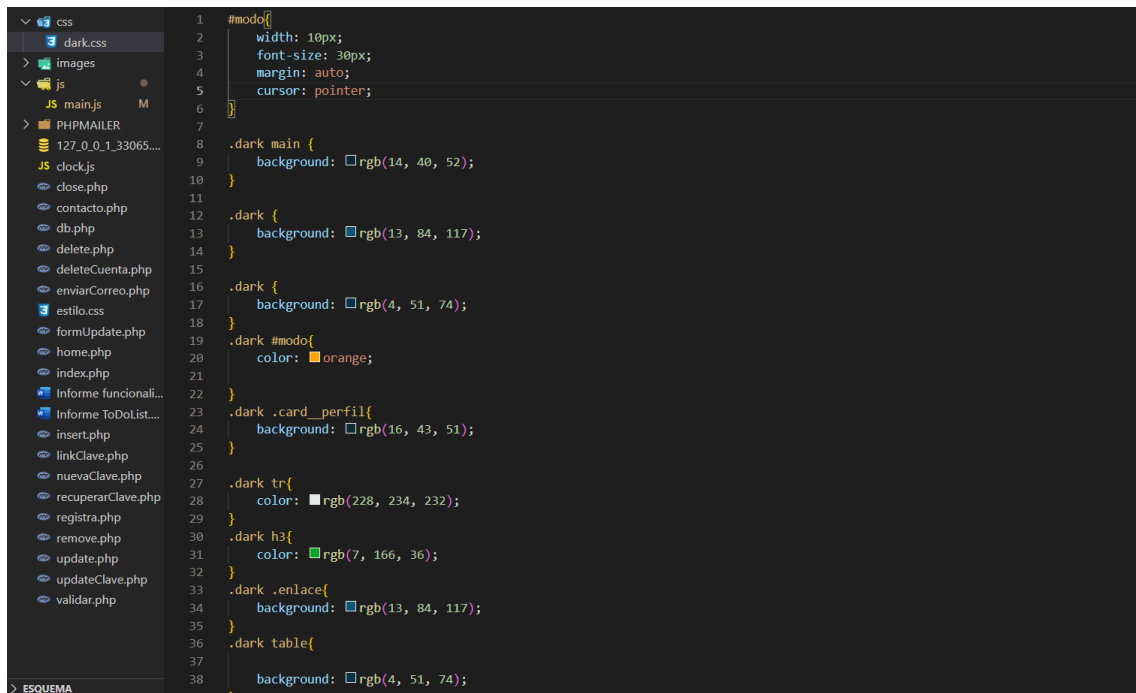


```
4 | let modo=document.getElementById("modo");
5 | let body=document.body;
6 |
7 | modo.addEventListener("click", function(){
8 |   let val=body.classList.toggle("dark")
9 |   localStorage.setItem("modo",val)
10 | })
11 |
12 | let valor=localStorage.getItem("modo")
13 |
14 | if (valor=="true") {
15 |   body.classList.add("dark")
16 | } else {
17 |   body.classList.remove("dark")
18 | }
```



Como vemos la funcionalidad es correcta, el cambio de tonalidad es notable significa que si funciona.

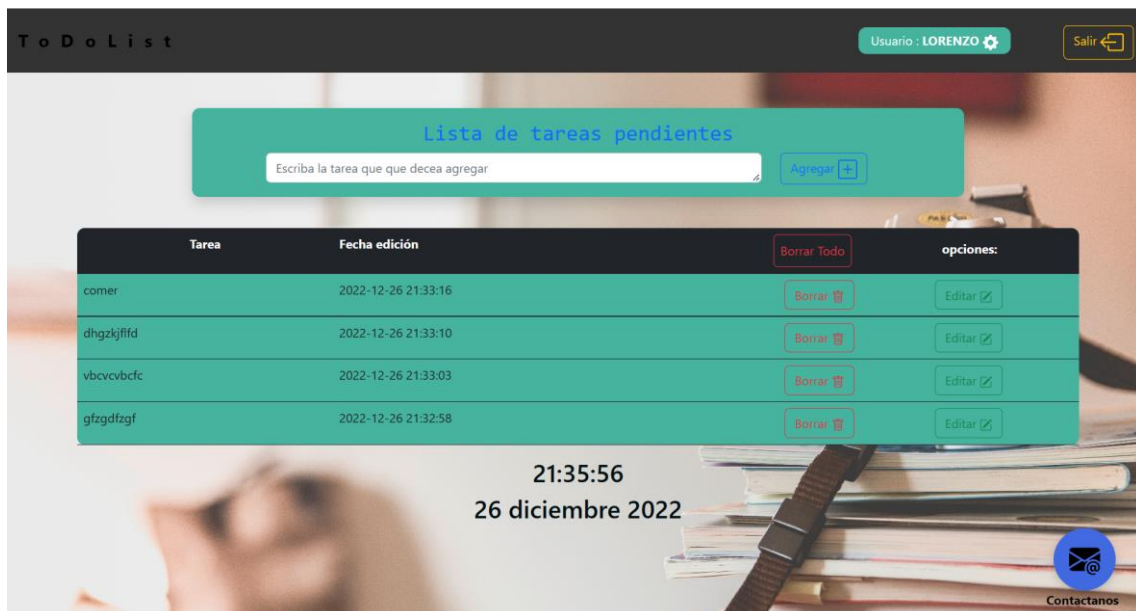
3. Seguidamente creamos una carpeta `css` en el que crearemos un archivo con el nombre `dark.css` de esta manera aplicaremos el modo oscuro a la zonas que se desea variando de un tono claro a uno oscuro.

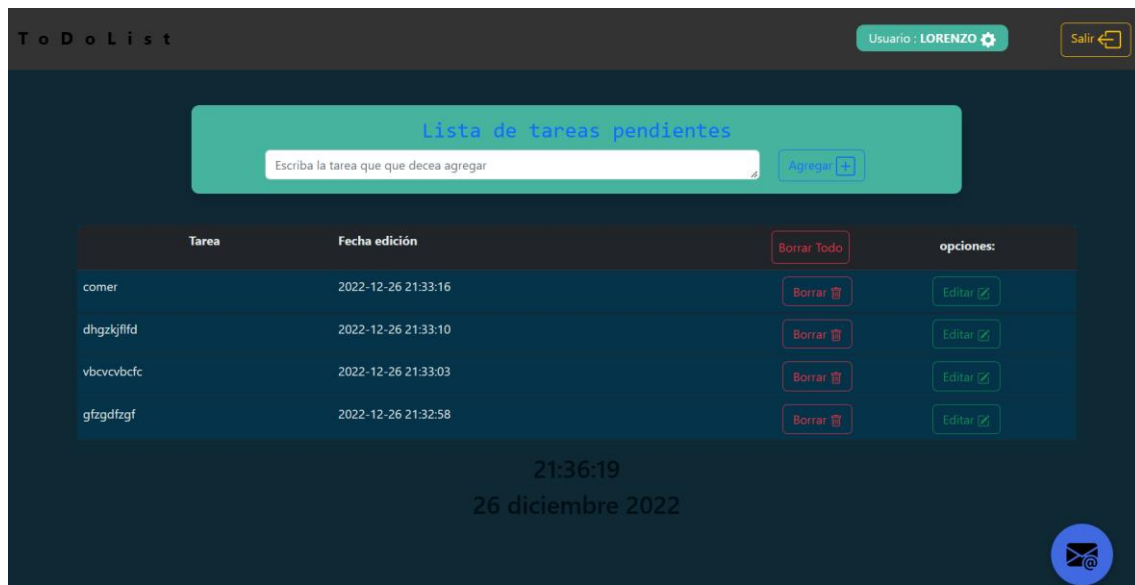


```

1 #modo{
2   width: 10px;
3   font-size: 30px;
4   margin: auto;
5   cursor: pointer;
6 }
7
8 .dark main {
9   background: #rgb(14, 40, 52);
10 }
11
12 .dark {
13   background: #rgb(13, 84, 117);
14 }
15
16 .dark {
17   background: #rgb(4, 51, 74);
18 }
19
20 .dark #modo{
21   color: orange;
22 }
23
24 .dark .card__perfil{
25   background: #rgb(16, 43, 51);
26 }
27
28 .dark tr{
29   color: #rgb(228, 234, 232);
30 }
31
32 .dark h3{
33   color: #rgb(7, 166, 36);
34 }
35
36 .dark .enlace{
37   background: #rgb(13, 84, 117);
38 }
39
40 .dark table{
41   background: #rgb(4, 51, 74);
42 }
  
```

Una vez aplicado las tonalidades que se desean a las partes que se quieren el resultado es este:





Como se puede observar funciona de forma correcta sin alterar las demás funcionalidades.