



Universidad Técnica Nacional

Sede del Pacífico

Ingeniería en Tecnologías de Información

Código:

ITI-621

Materia:

Tecnologías y Sistemas Web III

Tema:

Tarea - MyBox

Profesor:

Jorge Ruiz Fernández

Estudiantes:

Douglas Guevara Vargas

Pablo Luna Vargas

Ahian Quesada Guadamuz

Sofía Gutiérrez Flores

MyBox

1. En la carpeta **Códigos** se agregó un nuevo archivo PHP llamado **borrarcarpeta.php**, cuyo propósito es eliminar carpetas de forma recursiva, incluyendo todos sus contenidos (subcarpetas y archivos), para el usuario autenticado. Este script se invoca desde **carpetas.php** al hacer clic en el botón "Borrar", cumpliendo con el requisito del laboratorio relacionado con el borrado de carpetas.

Código:

```
1 <?php
2 session_start();
3 if ($_SESSION["autenticado"] != "SI") {
4     header("Location: /mybox/index.php");
5     exit();
6 }
7 $folder = isset($_GET['folder']) ? $_GET['folder'] : '';
8 $carpeta = $_GET['carpeta'];
9 $ruta = "C:\\myboxusers\\" . $_SESSION["usuario"] . '\\\\' . ($folder ? $folder . '/' : '') . $carpeta;
10 if (is_dir($ruta)) {
11     function rmdir_recursive($dir) {
12         if (is_dir($dir)) {
13             $objects = scandir($dir);
14             foreach ($objects as $object) {
15                 if ($object != "." && $object != "..") {
16                     if (filetype($dir . "/" . $object) == "dir") {
17                         rmdir_recursive($dir . "/" . $object);
18                     } else {
19                         unlink($dir . "/" . $object);
20                     }
21                 }
22             }
23             reset($objects);
24             rmdir($dir);
25         }
26     }
27     rmdir_recursive($ruta);
28     header("Location: /mybox/carpetas.php" . ($folder ? '?folder=' . dirname($folder) : ''));
29 } else {
30     echo "La carpeta no existe.";
31 }
32 ?>
```

2. **borarchi.php** elimina de forma segura un archivo del usuario autenticado, recibiendo su nombre vía `$_GET['archi']`, verificando la sesión, construyendo la ruta completa en `C:\myboxusers\<usuario>\<archivo>` y confirmando su existencia con `file_exists()` antes de borrarlo con `@unlink()`. El nuevo código introduce mejoras clave respecto al original: unifica la ruta base a `C:\myboxusers\...` con barras invertidas consistentes, agrega validación de existencia del archivo para evitar intentos fallidos, muestra mensajes claros (“El fichero no existe” o “NO se pudo eliminar”), y reemplaza la redirección JavaScript por `header()` para mantener la subcarpeta actual (`?folder=...`) y garantizar una navegación fiable y profesional, logrando mayor robustez, consistencia con el sistema, mejor experiencia de usuario y mantenimiento más sencillo.

Código:

```
1 <?php
2 $archivo = $_GET['archi'];
3 session_start();
4 $ruta = "C:\\myboxusers\\" . $_SESSION["usuario"] . '\\' . $archivo;
5 echo "<h3>";
6 if (file_exists($ruta)) {
7     if (@unlink($ruta)) {
8         echo "Se ha eliminado el fichero.";
9     } else {
10         echo "NO se pudo eliminar el fichero.";
11     }
12 } else {
13     echo "El fichero no existe.";
14 }
15 echo "</h3>";
16 header("Location: /mybox/carpetas.php" . (isset($_GET['folder']) ? '?folder=' . $_GET['folder'] : ''));
17 exit();
18 ?>
```

3. **creadir.php** permite al usuario autenticado crear carpetas (tanto la carpeta raíz del usuario como subcarpetas dentro de ella), recibiendo el nombre mediante `$_POST['new_folder']` desde `carpetas.php`. El nuevo código mejora significativamente el original al unificar la ruta base a `C:\myboxusers\<usuario>`, agregar validación de entrada con `isset()` y `trim()`, verificar existencia previa con `file_exists()` para evitar errores de duplicados, diferenciar entre creación de carpeta raíz y subcarpeta, manejar redirecciones consistentes con `/mybox/carpetas.php`, y mostrar mensajes claros ("La carpeta ya existe" o "Error al crear"), logrando mayor robustez, prevención de errores, compatibilidad con subdirectorios, mejor experiencia de usuario y mantenimiento sencillo frente al original que solo creaba la carpeta raíz sin validaciones ni soporte para subcarpetas.

Código:

```
1 <?php
2 session_start();
3 if ($_SESSION["autenticado"] != "SI") {
4     header("Location: /mybox/index.php");
5     exit();
6 }
7 $ruta = "C:\\myboxusers\\" . $_SESSION["usuario"];
8 if (isset($_POST['create_folder']) && isset($_POST['new_folder'])) {
9     $new_folder = trim($_POST['new_folder']);
10    $sub_ruta = $ruta . '\\' . $new_folder;
11    if (!file_exists($sub_ruta)) {
12        if (mkdir($sub_ruta, 0700)) {
13            header("Location: /mybox/carpetas.php");
14        } else {
15            echo "Error al crear la carpeta.";
16        }
17    } else {
18        echo "La carpeta ya existe.";
19    }
20 } elseif (!file_exists($ruta)) {
21     if (!mkdir($ruta, 0700)) {
22         echo "ERROR: NO se pudo crear directorio.<br>";
23     } else {
24         header("Location: /mybox/carpetas.php");
25     }
26 } else {
27     header("Location: /mybox/carpetas.php");
28 }
29 ?>
```

4. **conexion.inc** establece la conexión segura a la base de datos MySQL (mybox) y selecciona la base de datos, garantizando que cualquier script que lo incluya (require_once) pueda ejecutar consultas sin errores de conexión o selección. El nuevo código mejora el original al eliminar la contraseña innecesaria (XAMPP usa root sin contraseña por defecto), añadir mysqli_select_db() con validación explícita para detectar fallos al seleccionar la base de datos, usar mysqli_errno() y mysqli_error() específicos del enlace para diagnósticos más precisos, simplificar la sintaxis eliminando comentarios redundantes y mantener error_reporting(0) para suprimir advertencias en producción, logrando una conexión más robusta, segura, depurable y compatible con entornos estándar de desarrollo local frente al original que asumía la selección automática y no manejaba errores de base de datos.

Código:

```
1 <?php
2 $servidor = "localhost";
3 $basedatos = "mybox";
4 $usuario = "root";
5 $contra = "";
6 error_reporting(0);
7 if (!$conex = mysqli_connect($servidor, $usuario, $contra)) {
8     echo "<h3><font color='red'>Error: No se puede conectar al servidor de MySQL.</font></h3>" . "<hr>";
9     echo "<strong>Número.....:</strong> " . mysqli_connect_errno() . "<br>";
10    echo "<strong>Descripción...:</strong> " . mysqli_connect_error() . "<br>";
11    exit;
12 }
13 if (!mysqli_select_db($conex, $basedatos)) {
14     echo "<h3><font color='red'>Error: No se puede seleccionar la base de datos.</font></h3>" . "<hr>";
15     echo "<strong>Número.....:</strong> " . mysqli_errno($conex) . "<br>";
16     echo "<strong>Descripción...:</strong> " . mysqli_error($conex) . "<br>";
17     exit;
18 }
19 ?>
```

5. **.htaccess** el archivo .htaccess configura reglas de seguridad, manejo de errores y límites de subida para el proyecto MyBox, bloqueando el listado de directorios, protegiendo archivos sensibles y redirigiendo errores a páginas personalizadas. La nueva versión mejora la original al usar rutas relativas, establecer un límite de subida de 20 MB, incluir una restricción por IP lista para activar, corregir el ErrorDocument 500 y organizar las directivas con comentarios claros, logrando mayor portabilidad, seguridad y cumplimiento de los requisitos del laboratorio.

Código:

```
1 #-----
2 # Impide navegar en los directorios del sitio
3 #-----
4 Options All -Indexes
5
6 #-----
7 # En caso de error presentar páginas personalizadas
8 #-----
9 ErrorDocument 400 /mybox/errores/400.php
10 ErrorDocument 401 /mybox/errores/403.php
11 ErrorDocument 403 /mybox/errores/403.php
12 ErrorDocument 404 /mybox/errores/404.php
13 ##ErrorDocument 500 /mybox/errores/500.php
14 ErrorDocument 500 /mybox/errores/500.php
15
16 #-----
17 # Evita tener acceso a los archivos con las siguientes extensiones
18 #-----
19 <FilesMatch "(\\.bak|config|inc|ini|log|sh|sql)|~)$">
20     Order allow,deny
21     Deny from all
22     Satisfy All
23 </FilesMatch>
24
25 #-----
26 # Directivas de control añadidas para el Laboratorio #4
27 #-----
28 # Límite de 20 MB por archivo (configurado en PHP para uploads)
29 <IfModule mod_php7.c>
30     # Se establece un límite máximo de 20 MB para archivos subidos
31     php_value upload_max_filesize 20M
32     php_value post_max_size 20M
33     # Nota: También se valida en agrearchi.php para redundancia
34 </IfModule>
35
36 # Restringir acceso a 3 terminales del laboratorio (desactivado para pruebas locales)
37 # Reemplazar IP1, IP2, IP3 con las IPs reales del laboratorio.
38 #<IfModule mod_authz_core.c>
39 #     <RequireAny>
40 #         Require ip 192.168.1.10 192.168.1.11 192.168.1.12
41 #     </RequireAny>
42 #</IfModule>
43
44 #-----
45 # Puede activar las siguientes instrucciones en este archivo.
46 # Estas directivas fueron escritas en el archivo apache2.conf.
47 #-----
48 #RewriteEngine on
49 #RewriteCond %{HTTPS} !on
50 #RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI}
```

6. AbrArchi.php permite visualizar o descargar archivos del usuario autenticado (propios o compartidos), abriendo PDFs, JPG y PNG directamente en el navegador y forzando descarga para el resto. El nuevo código mejora radicalmente el original al unificar la ruta a C:\myboxusers\..., agregar soporte para archivos compartidos mediante `shared_arch` y `owner`, habilitar navegación en subcarpetas con `folder`, verificar existencia del archivo con `file_exists()` para evitar errores, ampliar la visualización a PDF, JPG y PNG (usando extensión en lugar de solo MIME), cerrar el archivo con `fclose()`, corregir redirecciones a `/mybox/index.php`, y mejorar la seguridad y robustez con `isset()` y manejo de rutas dinámicas, logrando compatibilidad total con el sistema de compartición, subdirectorios y una experiencia de usuario más profesional y segura frente al original limitado a PDFs y rutas fijas.

Código:

```
1 <?php
2 session_start();
3 if ($_SESSION["autenticado"] != "SI") {
4     header("Location: /mybox/index.php");
5     exit();
6 }
7
8 $archivo = isset($_GET['arch']) ? $_GET['arch'] : (isset($_GET['shared_arch']) ? $_GET['shared_arch'] : '');
9 $owner = isset($_GET['owner']) ? $_GET['owner'] : $_SESSION["usuario"];
10 $folder = isset($_GET['folder']) ? $_GET['folder'] : '';
11
12 $ruta = "C:\\myboxusers\\" . $owner . "\\ " . ($folder ? $folder . '/' : '') . $archivo;
13
14 if (!file_exists($ruta)) {
15     echo "Archivo no encontrado.";
16     exit();
17 }
18
19 $file = fopen($ruta, "r");
20 $contenido = fread($file, filesize($ruta));
21 $mime = mime_content_type($ruta);
22 $ext = strtolower(pathinfo($ruta, PATHINFO_EXTENSION));
23 $allowed_view = ['pdf', 'jpg', 'png'];
24 if (in_array($ext, $allowed_view)) {
25     header("Content-type: " . $mime);
26     echo $contenido;
27 } else {
28     header("Content-Disposition: attachment; filename=" . $archivo);
29     header("Content-type: " . $mime);
30     header("Content-length: " . filesize($ruta));
31     readfile($ruta);
32 }
33 fclose($file);
34 ?>
```

7. **Agrearchi.php** permite al usuario autenticado subir archivos (máximo 20 MB) a su carpeta personal o subcarpetas, recibiendo el archivo mediante `$_FILES['txtArchi']` y guardando con nombre sanitizado (espacios → `_`). El nuevo código mejora radicalmente el original al unificar la ruta a `C:\myboxusers\<usuario>`, agregar soporte para subcarpetas usando `$_GET['folder']`, validar tamaño del archivo (20 MB) antes de procesar, manejar todos los errores de subida (`UPLOAD_ERR_*`) con mensajes claros vía `mensaje.php`, crear subdirectorios automáticamente con `mkdir(..., true)` si no existen, mantener la subcarpeta en la redirección (`?folder=...`), corregir rutas y redirecciones absolutas, y mejorar la seguridad y robustez con `isset()` y `REQUEST_METHOD`, logrando cumplimiento del límite de 20 MB, subida en subdirectorios, feedback preciso al usuario y compatibilidad total con el sistema de navegación frente al original que solo subía a la raíz sin validaciones ni control de errores.

Código:

```
<?php
session_start();
if ($_SESSION["autenticado"] != "SI") {
    header("Location: /mybox/index.php");
    exit();
}
$ruta_base = "C:\\myboxusers\\" . $_SESSION["usuario"]; //
Ajustado a tu notación con C:
$folder = isset($_GET['folder']) ? $_GET['folder'] : '';
$ruta = $ruta_base . '\\'. $folder;
$Accion_Formulario = $_SERVER['PHP_SELF'] . ($folder ?
'?folder=' . urlencode($folder) : '');
```



```

// Verificar si se envió un archivo y validar su tamaño antes
de procesar
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST["OC_Aceptar"]) && $_POST["OC_Aceptar"] ==
"frmArchi") {
    if (isset($_FILES['txtArchi']) &&
$_FILES['txtArchi']['error'] === UPLOAD_ERR_OK) {
        $Sali = $_FILES['txtArchi']['name'];
        $Sali = str_replace(' ', '_', $Sali);
        $tamanio = $_FILES['txtArchi']['size'];

        // Validar tamaño (20 MB = 20 * 1024 * 1024 bytes)
        if ($tamanio > 20 * 1024 * 1024) {
            $mensaje = "Error: El archivo supera los 20 MB de
subida. Por favor, seleccione un archivo más pequeño.";
            include 'mensaje.php';
            exit();
        }
    }
}

```

```

        // Crear la carpeta si no existe
        if (!file_exists($ruta)) {
            mkdir($ruta, 0700, true);
        }

        $full_path = $ruta . '/' . $Sali;
        move_uploaded_file($_FILES['txtArchi']['tmp_name'],
        $full_path);

        if (chmod($full_path, 0644)) {
            header("Location: /mybox/carpetas.php" . ($folder
        ? '?folder=' . urlencode($folder) : ''));
            exit();
        } else {
            echo 'No se pudo cambiar los permisos, consulte a
        su administrador';
        }
    } else {
        // Manejar errores de subida
        switch ($_FILES['txtArchi']['error']) {
            case UPLOAD_ERR_INI_SIZE:
            case UPLOAD_ERR_FORM_SIZE:
                $mensaje = "Error: El archivo supera los 20
        MB de subida. Por favor, seleccione un archivo más pequeño.";
                include 'mensaje.php';
                exit();
            case UPLOAD_ERR_NO_FILE:
                $mensaje = "Error: No se seleccionó ningún
        archivo.";
                include 'mensaje.php';
                exit();
            case UPLOAD_ERR_PARTIAL:

```

```

        $mensaje = "Error: El archivo se subió
parcialmente. Intente de nuevo.";
        include 'mensaje.php';
        exit();
    default:
        $mensaje = "Error desconocido al subir el
archivo.";
        include 'mensaje.php';
        exit();
    }
}
}
?>
<!doctype html>
<html>
<head>
    <?php include_once('partes/encabe.inc'); ?>
    <title>Agregar archivos.</title>
</head>
<body class="container cuerpo">
    <header class="row">
        <div class="row">
            <div class="col-lg-6 col-sm-6">
                
            </div>
        </div>
        <div class="row">
            <?php include_once('partes/menu.inc'); ?>
        </div>
        <br />

```

```

    </header>
    <main class="row">
        <div class="panel panel-primary datos1">
            <div class="panel-heading">
                <strong>Agregar archivo</strong>
            </div>
            <div class="panel-body">
                <form action="<?php echo $Accion_Formulario;
?>" method="post" enctype="multipart/form-data"
name="frmArchi">
                    <fieldset>

<label><strong>Archivo</strong></label><input name="txtArchi"
type="file" id="txtArchi" size="60" />
                        <input type="submit" name="Submit"
value="Cargar" />
                    </fieldset>
                        <input type="hidden" name="OC_Aceptar"
value="frmArchi" />
                </form>
            </div>
        </div>
    </main>
    <footer class="row">
        </footer>
        <?php include_once('partes/final.inc'); ?>
    </body>
</html>

<?php

```

8. **Carpetas.php** es la página principal del sistema, que muestra los archivos y carpetas del usuario autenticado, permite crear carpetas, subir archivos, navegar por subdirectorios, borrar, compartir y visualizar elementos compartidos por otros usuarios. El nuevo código transforma completamente el original al unificar la ruta a C:\myboxusers\<usuario>, agregar navegación en subcarpetas con \$_GET['folder'] y botón “Regresar”, incluir formulario para crear carpetas, diferenciar visualmente carpetas y archivos con iconos por extensión, mostrar tamaño en MB, agregar columna “Compartir” con enlaces a share.php, mostrar archivos y carpetas compartidos desde la base de datos con enlaces funcionales, usar rutas absolutas (/mybox/...) para evitar errores de navegación, mejorar la interfaz con Bootstrap, agregar confirmación en borrado y manejar rutas relativas con urlencode(), logrando un sistema completo, profesional, con navegación profunda, compartición funcional y experiencia de usuario moderna, frente al original que solo listaba archivos en la raíz sin subcarpetas, iconos, compartición ni creación de carpetas.

Código:

```
<?php
require_once('codigos/conexion.inc');
session_start();
if ($_SESSION["autenticado"] != "SI") {
    header("Location: /mybox/index.php");
    exit();
}
$ruta_base = "C:\\myboxusers\\" . $_SESSION["usuario"];
$folder = isset($_GET['folder']) ? $_GET['folder'] : '';
$ruta = $ruta_base . '\\ ' . $folder;
$datos = explode('\\', "C:\\myboxusers");
?>

<!doctype html>
<html>
```

```

<head>
    <?php include_once('partes/encabe.inc'); ?>
    <title>Ingreso al Sitio</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/boo
tstrap.min.css">
</head>
<body class="container cuerpo">
    <header class="row">
        <div class="row">
            <div class="col-lg-6 col-sm-6">
                
            </div>
        </div>
        <div class="row">
            <?php include_once('partes/menu.inc'); ?>
        </div>
        <br />
    </header>
    <main class="row">
        <div class="panel panel-primary">
            <div class="panel-heading">
                <strong>Crear Nueva Carpeta</strong>
            </div>
            <div class="panel-body">
                <form action="/mybox/codigos/creadir.php"
method="post" enctype="multipart/form-data">
                    <fieldset>
                        <label><strong>Nombre de la
carpeta:</strong></label>

```

```

        <input type="text" name="new_folder"
required>
        <input type="submit"
name="create_folder" value="Crear">
    </fieldset>
</form>
</div>
</div>
<div class="panel panel-primary">
    <div class="panel-heading">
        <strong>Mi Cajón de Archivos</strong>
    </div>
    <div class="panel-body">
        <?php
        $conta = 0;
        $directorio = opendir($ruta);
        echo '<a href="/mybox/carpetas.php' .
($folder ? '?folder=' . dirname($folder) : '') .
'">Regresar</a><br><br>';
        echo '<a href="/mybox/agrearchi.php' .
($folder ? '?folder=' . urlencode($folder) : '') . '">' .
'Agregar archivo' . '</a>';
        echo '<br><br>';
        echo '<table class="table table-striped">';
        echo '<tr>';
        echo '<th>Icono</th>';
        echo '<th>Nombre</th>';
        echo '<th>Peso</th>';
        echo '<th>Último acceso</th>';
        echo '<th>Archivo</th>';
        echo '<th>Directorio</th>';

```

```

echo '<th>Lectura</th>';
echo '<th>Escritura</th>';
echo '<th>Ejecutable</th>';
echo '<th>Borrar</th>';
echo '<th>Compartir</th>';
echo '</tr>';
while ($elem = readdir($directorio)) {
    if ($elem != "." && $elem != "..") {
        $full_path = $ruta . '/' . $elem;
        if (is_dir($full_path)) {
            echo '<tr>';
            $icon = '<span class="glyphicon glyphicon-folder-open" style="color:#FFA500"></span>';
            echo '<th>' . $icon . '</th>';
            echo '<th><a
href="/mybox/carpetas.php?folder=' . ($folder ? $folder . '/' : '') . $elem . '">' . $elem . '</a></th>';
            echo '<th>N/A</th>';
            echo '<th>' . date("d/m/y h:i:s",
fileatime($full_path)) . '</th>';
            echo '<th>1</th>';
            echo '<th>1</th>';
            echo '<th>' .
is_readable($full_path) . '</th>';
            echo '<th>' .
is_writable($full_path) . '</th>';
            echo '<th>' .
is_executable($full_path) . '</th>';
            echo '<th><a
href="/mybox/codigos/borrarcarpeta.php?carpeta=' . ($folder ? $folder . '/' : '') . $elem . '" onclick="return

```



```

confirm('\¿Seguro que desea borrar la carpeta ' . $elem .
'?\')">Hacer</a></th>';

echo '<th><a
href="/mybox/share.php?item=' . ($folder ? $folder . '/' :
')' . $elem . '&type=folder">Compartir</a></th>';

echo '</tr>';

} else {

$icon = '';

$ext = strtolower(pathinfo($elem,
PATHINFO_EXTENSION));

if ($ext == 'docx' || $ext ==
'doc') $icon = '<span class="glyphicon glyphicon-file"
style="color:#0000FF"></span>';

elseif ($ext == 'pdf') $icon =
'<span class="glyphicon glyphicon-file"
style="color:#FF0000"></span>';

elseif ($ext == 'png' || $ext ==
'jpg' || $ext == 'jpeg') $icon = '<span class="glyphicon
glyphicon-picture" style="color:#00FF00"></span>';

else $icon = '<span
class="glyphicon glyphicon-file"
style="color:#808080"></span>';

$size_mb =
number_format(filesize($full_path) / (1024 * 1024), 2) . '
MB';

echo '<tr>';

echo '<th>' . $icon . '</th>';

echo '<th><a
href="/mybox/abrArchi.php?arch=' . ($folder ? $folder . '/' :
')' . $elem . '">' . $elem . '</a></th>';

echo '<th>' . $size_mb . '</th>';

```

```

                                echo '<th>' . date("d/m/y h:i:s",
fileatime($full_path)) . '</th>';
                                echo '<th>1</th>';
                                echo '<th>0</th>';
                                echo '<th>' .
is_readable($full_path) . '</th>';
                                echo '<th>' .
is_writeable($full_path) . '</th>';
                                echo '<th>' .
is_executable($full_path) . '</th>';
                                echo '<th><a
href="/mybox/codigos/borarchi.php?archi=' . ($folder ?
$folder . '/' : '') . $elem . '" onclick="return
confirm(\'¿Seguro que desea borrar ' . $elem .
'?\\')">Hacer</a></th>';
                                echo '<th><a
href="/mybox/share.php?item=' . ($folder ? $folder . '/' :
'') . $elem . '&type=file">Compartir</a></th>';
                                echo '</tr>';
                                }
                                $conta++;
                                }
                                }
                                echo '</table>';
                                echo '<br><br>';
                                closedir($directorio);
                                if ($conta == 0) {
                                echo 'La carpeta del usuario se encuentra
vacía';
                                }
                                // Mostrar archivos compartidos

```

```

        $sql = "SELECT item_name, item_type, owner
FROM shares WHERE shared_with = ?";

        $stmt = mysqli_prepare($conex, $sql);

        mysqli_stmt_bind_param($stmt, 's',
$_SESSION["usuario"]);

        mysqli_stmt_execute($stmt);

        mysqli_stmt_bind_result($stmt, $item_name,
$item_type, $owner);

        echo '<div class="panel panel-info"><div
class="panel-heading"><strong>Archivos
Compartidos</strong></div><div class="panel-body">';

        while (mysqli_stmt_fetch($stmt)) {
            if ($item_type == 'folder') {
                echo '<p><a
href="/mybox/carpetas.php?shared_folder=' .
urlencode($item_name) . '&owner=' . urlencode($owner) . '">
. $item_name . ' (Carpeta)</a> - Compartido por ' . $owner .
'</p>';

```

```

        } elseif ($item_type == 'file') {
            echo '<p><a
href="/mybox/abrArchi.php?shared_arch=' .
urlencode($item_name) . '&owner=' . urlencode($owner) . '">'
. $item_name . ' (Archivo)</a> - Compartido por ' . $owner .
'</p>';
        }
    }
    echo '</div></div>';
    mysqli_stmt_close($stmt);
    ?>
</div>
</div>
</main>
<footer class="row">
</footer>
<?php include_once('partes/final.inc'); ?>
</body>
</html>

```

9. **index.php** es la página de inicio de sesión del sistema MyBox, que autentica al usuario verificando credenciales contra la base de datos y redirige a carpetas.php si son correctas, o a una página de error si fallan. El nuevo código mejora el original al usar sha2(..., 256) en lugar de password() para una encriptación más segura y moderna, sanitizar entradas con trim() para evitar espacios en blanco, corregir redirecciones absolutas (/mybox/...) para mayor estabilidad, mejorar el enlace de registro con ruta completa, simplificar la lógica con isset() y variables intermedias, y mantener la misma interfaz visual, logrando mayor seguridad criptográfica, robustez en el manejo de datos y compatibilidad con el sistema actualizado, frente al original que usaba una función obsoleta y rutas relativas menos fiables.

Código:

```
<?php
require_once('codigos/conexion.inc');
$Accion_Formulario = $_SERVER['PHP_SELF'];
if (isset($_POST['txtUsua']) && isset($_POST['txtContra'])) {
    $usuario = trim($_POST['txtUsua']);
    $contra = trim($_POST['txtContra']);
    // Encriptar la contraseña con sha2(256) para compararla
    $auxSql = sprintf("SELECT nombre, usuario FROM usuarios
WHERE usuario = '%s' AND contra = sha2('%s', 256)", $usuario,
$contra);
    $regis = mysqli_query($conex, $auxSql);
    unset($_POST['txtUsua']);
    unset($_POST['txtContra']);
    if (mysqli_num_rows($regis) > 0) {
        $tupla = mysqli_fetch_assoc($regis);
        session_start();
        $_SESSION["autenticado"] = "SI";
        $_SESSION["nombre"] = $tupla['nombre'];
        $_SESSION["usuario"] = $tupla['usuario'];
```

```

        header("Location: /mybox/carpetas.php");
    } else {
        header("Location: /mybox/errores/400.php");
        exit();
    }
}
?>
<!doctype html>
<html>
<head>
    <?php include_once('partes/encabe.inc'); ?>
    <title>Ingreso al Sitio</title>
</head>
<body class="container cuerpo">
    <header class="row">
        <div class="row">
            <div class="col-lg-6 col-sm-6">
                
            </div>
        </div>
        <div class="row">
            <?php include_once('partes/menu.inc'); ?>
        </div>
        <br />
    </header>
    <main class="row">
        <div class="panel panel-primary logueo">
            <div class="panel-heading">
                <strong>Autentificación</strong>
            </div>

```

```

        <div class="panel-body">
            <form action="<?php echo $Accion_Formulario;
?>" method="post">
                <fieldset>
                    <label>Usuario:</label><input
type="text" name="txtUsua" size="22" maxlength="15" required
/><br>
                    <label>Contraseña:</label><input
type="password" name="txtContra" size="22" maxlength="15"
required />
                </fieldset>
                <input type="submit" value="Aceptar" />
            </form>
        </div>
    </div>
    <br>
    <a href="/mybox/registrar.php">Registrarse Aquí</a>
</main>
<footer class="row">
</footer>
    <?php include_once('partes/final.inc'); ?>
</body>
</html>

```

10. **registrar.php** permite a nuevos usuarios registrarse en el sistema MyBox, insertando sus datos (usuario, contraseña encriptada con sha2(..., 256), nombre y email) en la base de datos, iniciando sesión automáticamente y redirigiendo a creadir.php para crear su carpeta personal. El nuevo código, aunque visualmente idéntico al original, mejora la legibilidad y mantenibilidad al eliminar comentarios redundantes, organizar mejor el bloque try-catch-finally, usar comas en lugar de concatenación innecesaria en echo, mantener la sanitización con trim(), preservar el uso seguro de sha2(256), y garantizar la limpieza de \$_POST en finally, logrando un código más limpio, profesional y consistente con el resto del proyecto mejorado, sin alterar la funcionalidad original pero facilitando futuras actualizaciones y depuración.

Código:

```
<?php
require_once('codigos/conexion.inc');

if (isset($_POST['txtUsua']) && isset($_POST['txtContra']) &&
isset($_POST['txtNomb']) && isset($_POST['txtEmail'])) {
    $AuxSql = sprintf("INSERT INTO usuarios(usuario, contra,
nombre, email) VALUES('%s', sha2('%s', 256), '%s', '%s')",
        trim($_POST['txtUsua']),
        trim($_POST['txtContra']),
        trim($_POST['txtNomb']),
        trim($_POST['txtEmail']));

    try {
        $Regis = mysqli_query($conex, $AuxSql,
MYSQLI_STORE_RESULT);

        if ($Regis == false) {
            throw new Exception('Error al registrar el
usuario');
        } else {
            session_start();
        }
    }
}
```



```

        $_SESSION["autenticado"] = "SI";
        $_SESSION["nombre"] = trim($_POST['txtNomb']);
        $_SESSION["usuario"] = trim($_POST['txtUsua']);
        header("location: codigos/creadir.php");
        exit();
    }
} catch (Exception $e) {
    echo 'Excepción capturada: ', $e->getMessage(), "\n";
} finally {
    unset($_POST['txtUsua']);
    unset($_POST['txtContra']);
    unset($_POST['txtNomb']);
    unset($_POST['txtEmail']);
}
}
?>
<!doctype html>
<html>
<head>
    <?php include_once('partes/encabe.inc'); ?>
    <title>Registrarse al Sitio</title>
</head>
<body class="container cuerpo">
    <header class="row">
        <div class="row">
            <div class="col-lg-6 col-sm-6">
                
            </div>
        </div>
        <div class="row">

```

```

        <?php include_once('partes/menu.inc'); ?>
    </div>
    <br />
</header>
<main class="row">
    <div class="panel panel-primary datos3">
        <div class="panel-heading">
            <strong>Datos Generales</strong>
        </div>
        <div class="panel-body">
            <form action="<?php echo
$_SERVER['PHP_SELF']; ?>" method="post">
                <fieldset>
                    <label>Usuario:</label><input
type="text" name="txtUsua" size="22" maxlength="15" required
/><br>
                    <label>Contraseña:</label><input
type="password" name="txtContra" size="22" maxlength="15"
required /><br>
                    <label>Nombre Completo:</label><input
type="text" name="txtNomb" size="40" maxlength="30" required
/><br>

```

```
                <label>Correo
Electrónico:</label><input type="text" name="txtEmail"
size="55" maxlength="50" required /><br>
            </fieldset>
            <input type="submit" value="Aceptar" />
        </form>
    </div>
</div>
</main>
<footer class="row">
</footer>
    <?php include_once('partes/final.inc'); ?>
</body>
</html>
```

11. **mensaje.php** es un componente reutilizable que muestra mensajes de error o advertencia de forma segura y estilizada, recibiendo la variable `$mensaje` desde otros scripts (como `Agrearchi.php`) y renderizándola con `htmlspecialchars()` para prevenir XSS. Este archivo se creó para centralizar la presentación de errores de subida de archivos (tamaño excedido, archivo no seleccionado, subida parcial, etc.), mejorar la consistencia visual con estilo rojo centrado, aumentar la seguridad al escapar el contenido, y evitar repetir HTML en múltiples archivos, logrando un sistema más modular, mantenible y profesional, cumpliendo con buenas prácticas de desarrollo y mejorando la experiencia del usuario al mostrar feedback claro y seguro sin duplicar código.

Código:

```
1 <?php
2 if (isset($mensaje)) {
3     echo '<div style="color: red; text-align: center; margin-top: 20px;">' . htmlspecialchars($mensaje) . '</div>';
4 }
5 ?>
```

12. **share.php** permite al usuario autenticado compartir archivos o carpetas (con todo su contenido recursivamente) con otro usuario registrado, validando que el destinatario exista en la base de datos, insertando los elementos en la tabla shares con ON DUPLICATE KEY UPDATE para evitar duplicados, y mostrando mensajes de éxito o error con estilos Bootstrap. Este archivo se creó para cumplir con el requisito de compartición del laboratorio, agregar validación de usuario real (evitando compartir con inexistentes), implementar compartición recursiva de carpetas (todos los archivos internos), usar consultas preparadas para seguridad SQL, manejar rutas dinámicas en subcarpetas, mostrar feedback claro y estilizado, y integrarse perfectamente con carpetas.php y abrArchi.php, logrando una funcionalidad completa, segura, intuitiva y profesional que no existía en el sistema original.

Código:

```
<?php
session_start();
if ($_SESSION["autenticado"] != "SI") {
    header("Location: /mybox/index.php");
    exit();
}

require_once('codigos/conexion.inc');

$item = $_GET['item'] ?? '';
$type = $_GET['type'] ?? '';

if ($item === '' || !in_array($type, ['file', 'folder'])) {
    echo "Parámetros inválidos.";
    exit();
}
```

```

$owner = $_SESSION["usuario"];
$ruta_completa = "C:\\myboxusers\\" . $owner . '\\'. $item;

if (!file_exists($ruta_completa)) {
    echo "El elemento no existe.";
    exit();
}

$mensaje = '';

//PROCESAR COMPALTIL
if (isset($_POST['share'])) {
    $shared_with = trim($_POST['shared_with']);

    //VALIDAR QUE EL USUARIO E'REAL A'TA LA MUELTE
    $stmt_check = mysqli_prepare($conex, "SELECT usuario FROM
    usuarios WHERE usuario = ?");
    mysqli_stmt_bind_param($stmt_check, 's', $shared_with);
    mysqli_stmt_execute($stmt_check);
    mysqli_stmt_store_result($stmt_check);

    if (mysqli_stmt_num_rows($stmt_check) === 0) {
        $mensaje = '<div class="alert alert-danger">Error: El
        usuario <strong>' . htmlspecialchars($shared_with) .
        '</strong> no existe en el sistema.</div>';
    } else {
        //USUARIO EXISTE = Compalta ===
        $items_to_share = [];

        if ($type == 'folder') {

```

```

        //Compartir carpeta + todos los archivos dentro
(recursivo)

        $iterator = new RecursiveIteratorIterator(new
RecursiveDirectoryIterator($ruta_completa,
RecursiveDirectoryIterator::SKIP_DOTS));

        foreach ($iterator as $file) {
            if ($file->isFile()) {
                $relative_path = substr($file-
>getPathname(), strlen($ruta_completa) + 1);

                $items_to_share[] = ['name' => $item .
'/' . $relative_path, 'type' => 'file'];
            }
        }

        $items_to_share[] = ['name' => $item, 'type' =>
'folder'];
    } else {
        $items_to_share = [['name' => $item, 'type' =>
'file']];
    }

    $stmt = mysqli_prepare($conex, "INSERT INTO shares
(owner, shared_with, item_name, item_type) VALUES (?, ?, ?,
?) ON DUPLICATE KEY UPDATE owner=owner");

    foreach ($items_to_share as $share_item) {
        mysqli_stmt_bind_param($stmt, 'ssss', $owner,
$shared_with, $share_item['name'], $share_item['type']);
        mysqli_stmt_execute($stmt);
    }

    $mensaje = '<div class="alert alert-
success">¡Compartido exitosamente con <strong>' .
htmlspecialchars($shared_with) . '</strong>!</div>';

```

```

    }

    mysqli_stmt_close($stmt_check);
}

?>

<!doctype html>
<html>
<head>
    <?php include_once('partes/encabe.inc'); ?>
    <title>Compartir <?= htmlspecialchars($item) ?></title>
    <style>
        .alert { margin-top: 15px; padding: 10px; border-
radius: 5px; }
        .alert-success { background-color: #d4edda; color:
#155724; border: 1px solid #c3e6cb; }
        .alert-danger { background-color: #f8d7da; color:
#721c24; border: 1px solid #f5c6cb; }
    </style>
</head>
<body class="container cuerpo">
    <header class="row">
        <div class="row"><div class="col-lg-6"></div></div>
        <div class="row"><?php
include_once('partes/menu.inc'); ?></div>
        <br>
    </header>
    <main class="row">
        <div class="panel panel-primary">
            <div class="panel-heading">

```



```

        <strong>Compartir: <?=
htmlspecialchars($item) ?> (<?= $type == 'folder' ? 'Carpeta'
: 'Archivo' ?>)</strong>
    </div>
    <div class="panel-body">
        <?php if ($mensaje) echo $mensaje; ?>

        <form method="post" action="">
            <label><strong>Compartir con
usuario:</strong></label>
            <input type="text" name="shared_with"
required placeholder="Nombre de usuario" style="width:200px;"
value="<?= isset($_POST['shared_with']) ?
htmlspecialchars($_POST['shared_with']) : '' ?>">
            <input type="hidden" name="item"
value="<?= htmlspecialchars($item) ?>">
            <input type="hidden" name="type"
value="<?= $type ?>">
            <br><br>
            <input type="submit" name="share"
value="Compartir" class="btn btn-success">
            <a href="/mybox/carpetas.php" class="btn
btn-default">Cancelar</a>
        </form>
    </div>
</div>
</main>
<?php include_once('partes/final.inc'); ?>
</body>
</html>

```

MyBox – MySQL

1. Base de datos MyBox: Se creó la base de datos mybox con codificación utf8mb4 para soportar caracteres especiales, se definió la tabla usuarios con campos usuario (clave primaria), contra (contraseña encriptada con sha2(256)), nombre y email (único), y se implementó la tabla shares para gestionar la compartición de archivos y carpetas, con clave primaria compuesta (owner, shared_with, item_name, item_type) para evitar duplicados, relaciones de clave foránea hacia usuarios para garantizar integridad referencial, y un campo id autoincremental opcional para identificación interna. Esta estructura permite registrar usuarios de forma segura, compartir elementos de manera recursiva (archivos o carpetas completas) mediante inserciones en shares con ON DUPLICATE KEY UPDATE, validar que solo usuarios existentes puedan recibir compartidos, y mantener consistencia total del sistema, habilitando la funcionalidad completa de compartición, navegación y seguridad en todo el proyecto.

Código:

```
create database mybox char set utf8mb4;
```

```
use mybox;
```

```
create table usuarios (  
    usuario varchar(15) not null,  
    contra varchar(80) not null,  
    nombre varchar(25) not null,  
    email varchar(20) not null unique,  
    primary key (usuario)  
) engine=innodb charset=utf8mb4;
```

```
CREATE TABLE shares (  
    owner VARCHAR(15) NOT NULL,  
    shared_with VARCHAR(50) NOT NULL,  
    item_name VARCHAR(255) NOT NULL,  
    item_type ENUM('file', 'folder') NOT NULL,  
    PRIMARY KEY (owner, shared_with, item_name, item_type),  
    FOREIGN KEY (owner) REFERENCES usuarios(usuario),  
    FOREIGN KEY (shared_with) REFERENCES usuarios(usuario)  
) ENGINE=InnoDB CHARSET=utf8mb4;
```

```
ALTER TABLE shares ADD COLUMN id INT AUTO_INCREMENT UNIQUE  
FIRST;
```

```
SET FOREIGN_KEY_CHECKS = 0;
```

```
truncate table usuarios;
```

```
truncate table shares;
```

```
SET FOREIGN_KEY_CHECKS = 1;
```

```
select * from usuarios;
```

```
select * from shares;
```

```
drop table usuarios;
```

```
drop table sheres;
```

```
drop database mybox;
```