

# Disco duro virtual en PHP

Daniel Alfonso Gómara Ruiz de Galarreta

09/01/2019

Desarrollo Web en Entorno Servidor

Desarrollo de Aplicaciones Web



## RESUMEN

El presente proyecto versa sobre la creación de un disco duro virtual en PHP. Se recogen una serie de funciones que permiten guardar, borrar y descargar los ficheros de un conjunto de usuarios. Cada usuario gestiona sus propios ficheros independientemente de los demás.

## ÍNDICE

Resumen.....	2
Índice.....	3
Índice de figuras .....	4
Disco duro virtual en php.....	5
BIBLIOGRAFÍA .....	21

---

## ÍNDICE DE FIGURAS

• Figura 1. Contenido del archivo “./seguridad/ficheros/datosBD.php”.....	6
• Figura 2. Contenido del archivo “./seguridad/ficheros/funciones.php”.....	7
• Figura 3. Contenido del archivo “./seguridad/ficheros/validarSesion-s.php” (líneas 1-22).....	7
• Figura 4. Contenido del archivo “./seguridad/ficheros/validarSesion-s.php” (líneas 22-46).....	8
• Figura 5. Contenido del archivo “./seguridad/ficheros/validarSesion-s.php” (líneas 46-55).....	8
• Figura 6. Contenido del archivo “./www/paginaWeb/src/cerrarSesion.php”.....	9
• Figura 7. Contenido del archivo “./www/paginaWeb/src/subirFicheros.php” (líneas 1-19).....	9
• Figura 8. Contenido del archivo “./www/paginaWeb/src/subirFicheros.php” (líneas 19-55).....	10
• Figura 9. Contenido del archivo “./www/paginaWeb/src/subirFicheros.php” (líneas 55-95).....	11
• Figura 9. Contenido del archivo “./www/paginaWeb/src/subirFicheros.php” (líneas 95-102).....	11
• Figura 10. Contenido del archivo “./www/paginaWeb/src/descargarFicheros.php” (líneas 1-18).....	12
• Figura 11. Contenido del archivo “./www/paginaWeb/src/descargarFicheros.php” (líneas 18-45).....	12
• Figura 12. Contenido del archivo “./www/paginaWeb/src/borrarFicheros.php” (líneas 1-19).....	13
• Figura 13. Contenido del archivo “./www/paginaWeb/src/borrarFicheros.php” (líneas 19-52).....	13
• Figura 14. Contenido del archivo “./www/paginaWeb/src/borrarFicheros.php” (líneas 52-89).....	14
• Figura 15. Contenido del archivo “./seguridad/ficheros/index-s.php”.....	15
• Figura 16. Contenido del archivo “./www/paginaWeb/index.php”.....	15
• Figura 17. Contenido del archivo “./seguridad/ficheros/discoDuro-s.php”.....	16
• Figura 18. Contenido del archivo “./www/paginaWeb/discoDuro.php” (líneas 1-37).....	16
• Figura 19. Contenido del archivo “./www/paginaWeb/discoDuro.php” (líneas 37-69).....	17
• Figura 20. Contenido del archivo “./www/paginaWeb/discoDuro.php” (líneas 70-118).....	17
• Figura 21. Contenido del archivo “./www/paginaWeb/discoDuro.php” (líneas 118-128).....	18
• Figura 22. Contenido del archivo “./seguridad/ficheros/validarSesion-s.php”.....	18
• Figura 23. Vista de la página de login.....	20
• Figura 24. Vista de la página del disco duro virtual.....	20

## DISCO DURO VIRTUAL EN PHP

### 1. RESUMEN

El disco duro permitirá hacer login con un usuario previamente creado. Cada usuario podrá guardar, borrar y descargar ficheros de forma independiente al resto de usuarios. Cada usuario se compone entonces de cuatro componentes: nombre de usuario, nombre común, contraseña y cuota de almacenamiento.

### 2. MANUAL TÉCNICO

#### 2.1. FLUJO DE SCRIPTS

- Se accede a la página principal “index.php”.
- Antes de cargar el código HTML de “index.php” se comprueba que haya una sesión activa. Si la hay se redirecciona a la página “discoDuro.php”, si no la hay carga el código HTML de la página.
- El usuario introduce sus credenciales y se validan con la información que hay en la base de datos. Si la validación es correcta se crea una sesión para dicho usuario, si la validación es errónea se vuelve a cargar “index.php” con un mensaje indicando el error.
- Antes de cargar el código HTML de “discoDuro.php” se comprueba que haya una sesión activa. Si la hay se carga el código HTML de la página, si no la hay se redirecciona a la página “index.php”.
- El usuario podrá entonces subir archivos al disco duro virtual. Si el total de los archivos es igual o menor que la cuota disponible para ese usuario se subirán los archivos correctamente y se reducirá la cuota, si el total de los archivos es mayor que la cuota se volverá a cargar “discoDuro.php” con un mensaje indicando el error.
- El usuario podrá entonces descargar los archivos subidos desde el disco duro virtual. Si no hay ningún archivo, se mostrará un mensaje indicándolo.

- El usuario podrá también borrar los archivos subidos en el disco duro virtual. Se aumentará la cuota para ese usuario en la misma cantidad que el tamaño del archivo borrado.
- El usuario podrá cerrar la sesión en cualquier momento, volviendo a la página de inicio "index.php".

## 2.2. DESCRIPCIÓN TÉCNICA DE SCRIPTS

### 2.2.1. DEFINIR LAS CREDENCIALES DE LA BASE DE DATOS

Para poder acceder a la base de datos necesitamos definir unas constantes con las credenciales: dirección IP, usuario, clave y nombre de la base de datos. También se necesita definir otra constante con la ruta del directorio en la que se guardarán los archivos subidos.

```
1  <!-- Credenciales de la base de datos -->
2
3  <?php
4      define("IP", "127.0.0.1");
5      define("USUARIO", "discoduro");
6      define("CLAVE", "discoduro");
7      define("BD", "discoduro");
8      define("__ALMACEN__", "../..../seguridad/ficheros/almacen/");
9  ?>
10
```

Figura 1. Contenido del archivo "./seguridad/ficheros/datosBD.php".

Fuente: Elaboración propia.

### 2.1.2 FUNCIONES PARA EL MANEJO DE SESIONES

Las páginas de inicio y del disco duro necesitan de dos funciones para controlar las sesiones: una para crear y mantener abiertas las sesiones y otra para comprobar haya una sesión activa.

```

1  <!-- Funciones de la páginas -->
2
3  <?php
4  function inicioSesion() {
5      session_name("SESION");
6      session_cache_limiter("nocache");
7      session_start();
8  }
9
10 function validado(&$usuario) {
11     $validado = false;
12     if (isset($_SESSION["validado"]) && $_SESSION["validado"]) {
13         $validado = true;
14         $usuario = $_SESSION["usuario"];
15     }
16     return $validado;
17 }
18 ?>
19

```

Figura 2. Contenido del archivo “./seguridad/ficheros/funciones.php”.

Fuente: Elaboración propia.

### 2.1.3 VALIDACIÓN DEL USUARIO

Para validar el usuario se ha de seguir una determinada serie de pasos. Primero comprobamos que recibimos mediante el método POST el nombre de usuario y la clave desde el formulario de login, además de que ambos no estén vacíos ni superen los 20 caracteres. En el caso de que no se cumplan estas condiciones, se redirecciona a “index.php” con el correspondiente mensaje de error.

```

1  <!-- Validación del usuario -->
2
3  <?php
4  $usuario = "";
5  if (!isset($_POST["usuario"])) {
6      header("Location: ../index.php?mensaje=".urlencode("ERROR DE SERVIDOR: No se ha podido recibir el nombre de usuario."));
7      exit;
8  }
9  $usuario = strip_tags(trim($_POST["usuario"]));
10
11 $clave = "";
12 if (!isset($_POST["clave"])) {
13     header("Location: ../index.php?mensaje=".urlencode("ERROR DE SERVIDOR: No se ha podido recibir la contraseña."));
14     exit;
15 }
16 $clave = strip_tags(trim($_POST["clave"]));
17
18 if (empty($usuario) || strlen($usuario)>20 || empty($clave) || strlen($clave)>20) {
19     header("Location: ../index.php?mensaje=".urlencode("El usuario no existe o la contraseña no es válida."));
20     exit;
21 }
22

```

Figura 3. Contenido del archivo “./seguridad/ficheros/validarSesion-s.php” (líneas 1-22).

Fuente: Elaboración propia.

Una vez comprobado esto, creamos un canal con la base de datos para realizar una consulta preparada. Pediremos la clave que corresponda al nombre de usuario introducido, que se nos devolverá en formato encriptado. Para comparar la clave de la base de datos con la que nos ha mandado el usuario, debemos utilizar la función “password\_verify()”. En el caso de que la clave introducida por el usuario sea incorrecta, se redirecciona a “index.php” con el correspondiente mensaje de error.

```

22
23     $canal = @mysqli_connect(IP, USUARIO, CLAVE, BD);
24     if (!$canal) {
25         echo "Ha ocurrido el error: ".mysqli_connect_errno()." ".mysqli_connect_error()."<br />";
26         exit;
27     }
28     mysqli_set_charset($canal, "utf8");
29
30     $sql = "SELECT clave FROM usuarios WHERE usuario = ?";
31     $consulta = mysqli_prepare($canal, $sql);
32     if (!$consulta) {
33         echo "Ha ocurrido el error: ".mysqli_errno($canal)." ".mysqli_error($canal)."<br />";
34         exit;
35     }
36     mysqli_stmt_bind_param($consulta, "s", $usuario);
37
38     mysqli_stmt_execute($consulta);
39     mysqli_stmt_bind_result($consulta, $clave);
40     mysqli_stmt_fetch($consulta);
41
42     if (!password_verify($clave, $clave)) {
43         header("Location: ../../index.php?mensaje=".urlencode("El usuario no existe o la contraseña no es válida."));
44         exit;
45     }
46

```

Figura 4. Contenido del archivo “./seguridad/ficheros/validarSesion-s.php” (líneas 22-46).

Fuente: Elaboración propia.

Finalmente, cerramos y destruimos la consulta, cerramos el canal, creamos una sesión con la función “inicioSesion()” (ver figura 2) y guardamos dos valores dentro de esta sesión: que ha sido validada correctamente y el nombre de usuario.

```

46
47     mysqli_stmt_close($consulta);
48     unset($consulta);
49     mysqli_close($canal);
50
51     inicioSesion();
52     $_SESSION["validado"] = true;
53     $_SESSION["usuario"] = $usuario;
54     ?>
55

```

Figura 5. Contenido del archivo “./seguridad/ficheros/validarSesion-s.php” (líneas 46-55).

Fuente: Elaboración propia.



## 2.1.4 CERRAR SESIÓN

A la hora de cerrar la sesión, tenemos que asegurarnos de haya una sesión abierta (por lo que si no la hubiera, se crea una sesión “pelele” para que no den error las siguientes instrucciones). Destruimos todos los datos almacenados en la sesión, así como la variable asignada a ella. Finalmente, redirigimos a la página principal junto con un mensaje confirmando que la sesión ya no está activa.

```

1  <!-- Cerrar la sesión actual y volver al index -->
2
3  <?php
4      include "../../seguridad/ficheros/funciones.php";
5      inicioSesion();
6      session_destroy();
7      unset($_SESSION);
8      header("Location: ../index.php?mensaje=".urlencode("Se ha cerrado la sesión."));
9      exit;
10 ?>
11

```

Figura 6. Contenido del archivo “./www/paginaWeb/src/cerrarSesion.php”.

Fuente: Elaboración propia.

## 2.1.5 SUBIDA DE FICHEROS

Para subir ficheros al disco duro virtual, se han de seguir una serie de pasos y comprobaciones. Lo primero de todo es comprobar que se ha podido recibir el array de ficheros desde el formulario. Si es así, guardamos en una variable el número de ficheros que se intentan subir. También guardamos el nombre de usuario en una variable local. Si en cualquiera de los dos casos se produjera un error, se volvería a cargar la página mostrando un mensaje con el correspondiente error.

```

1  <!-- Subir los ficheros introducidos en el formulario -->
2
3  <?php
4      include "../../seguridad/ficheros/datosBD.php";
5      include "../../seguridad/ficheros/funciones.php";
6      inicioSesion();
7
8      if (!isset($_FILES["ficheros"]) || !is_array($_FILES["ficheros"]["name"])) {
9          header("Location: ../discoDuro.php?mensaje=".urlencode("ERROR DE SERVIDOR: No se han podido recibir los ficheros."));
10         exit;
11     }
12     $intentosSubida = count($_FILES["ficheros"]["name"]);
13
14     if (!isset($_SESSION["usuario"])) {
15         header("Location: ../discoDuro.php?mensaje=".urlencode("ERROR DE SERVIDOR: No se han podido recibir el usuario."));
16         exit;
17     }
18     $usuario = $_SESSION["usuario"];
19

```

Figura 7. Contenido del archivo “./www/paginaWeb/src/subirFicheros.php” (líneas 1-19).

Fuente: Elaboración propia.

Creamos un canal con la base de datos para realizar las consultas preparadas. La primera consulta la ejecutaremos nada más declararla, en la cual pediremos a la base de datos la cuota correspondiente al usuario que tiene la sesión activa. El valor lo guardaremos en una variable local. Después prepararemos dos consultas más pero sin ejecutarlas todavía, una para insertar en la tabla “ficheros” los datos de un fichero y otra consulta para actualizar la cuota.

```

19     $canal = @mysqli_connect(IP, USUARIO, CLAVE, BD);
20
21     if (!$canal) {
22         echo "Ha ocurrido el error: ".mysqli_connect_errno(). " ".mysqli_connect_error(). "<br />";
23         exit;
24     }
25     mysqli_set_charset($canal, "utf8");
26
27     $sql = "SELECT cuota FROM usuarios WHERE usuario = ?";
28     $consulta = mysqli_prepare($canal, $sql);
29     if (!$consulta) {
30         echo "Ha ocurrido el error: ".mysqli_errno($canal). " ".mysqli_error($canal). "<br />";
31         exit;
32     }
33     mysqli_stmt_bind_param($consulta, "s", $usuario_);
34     $usuario_ = $usuario;
35     mysqli_stmt_execute($consulta);
36     mysqli_stmt_bind_result($consulta, $cuota);
37     mysqli_stmt_fetch($consulta);
38     unset($consulta);
39
40     $sql1 = "INSERT INTO ficheros (id, nombre, tamaño, tipo, usuario) values (?, ?, ?, ?, ?)";
41     $consulta1 = mysqli_prepare($canal, $sql1);
42     if (!$consulta1) {
43         echo "Ha ocurrido el error: ".mysqli_errno($canal). " ".mysqli_error($canal). "<br />";
44         exit;
45     }
46     mysqli_stmt_bind_param($consulta1, "ssiss", $id_, $nombre_, $tamanyo_, $tipo_, $usuario_);
47
48     $sql2 = "UPDATE usuarios SET cuota = ? WHERE usuario = ?";
49     $consulta2 = mysqli_prepare($canal, $sql2);
50     if (!$consulta2) {
51         echo "Ha ocurrido el error: ".mysqli_errno($canal). " ".mysqli_error($canal). "<br />";
52         exit;
53     }
54     mysqli_stmt_bind_param($consulta2, "ss", $cuota_, $usuario_);
55

```

Figura 8. Contenido del archivo “./www/paginaWeb/src/subirFicheros.php” (líneas 19-55).

Fuente: Elaboración propia.

Una vez tengamos todo preparado, procedemos a comprobar los ficheros del array uno por uno. En el segmento “error” del array de ficheros encontramos un código de error que se genera con la subida de ficheros. Entonces si el código de error es:

- “UPLOAD\_ERR\_OK”: El fichero se puede subir sin errores. Comprobamos que el tamaño del fichero, sumado al tamaño total de los ficheros que se han subido anteriormente con éxito dentro del mismo lote, no excede la cuota límite del usuario. De ser así, se cancela la subida de ese archivo y de los siguientes y se muestra un mensaje de error. En el caso de que no supere la cuota, se mueve el archivo al “almacén”, se guardan sus propiedades en variables locales y se ejecutan las dos consultas que teníamos preparadas anteriormente.
- “UPLOAD\_ERR\_NO\_FILE”: Se ha intentado subir un array de ficheros vacío. Por consiguiente, se mostrará un mensaje de error indicando esto.

- “UPLOAD\_ERR\_INI\_SIZE” y “UPLOAD\_ERR\_FORM\_SIZE”: El fichero en cuestión supera la cuota disponible por sí mismo, sin necesidad de ser sumada al tamaño total del lote de subida. Se cancela la subida y se muestra un mensaje de error apropiado.
- En el caso de que el código de error que devuelva el array de ficheros no se incluya entre los anteriormente mencionados, se mostrará un mensaje genérico indicando el fallo como “desconocido”.

```

55
56 $mensaje = "";
57 $tamanyoTotal = 0;
58 for ($i = 0; $i < $intentosSubida; $i++) {
59     switch ($_FILES["ficheros"]["error"][$i]) {
60         case UPLOAD_ERR_OK:
61             $id = uniqid("", true);
62             $ficheroSubido = __ALMACEN__.$id;
63             if ($tamanyoTotal <= $cuota) {
64                 $tamanyoTotal = $tamanyoTotal + $_FILES["ficheros"]["size"][$i];
65                 $cuota = $cuota - $_FILES["ficheros"]["size"][$i];
66                 if (move_uploaded_file($_FILES["ficheros"]["tmp_name"][$i], $ficheroSubido)) {
67                     // $mensaje .= "Fichero \"".$_FILES["ficheros"]["name"][$i]."\n subido con éxito.";
68                     $nombre = basename($_FILES["ficheros"]["name"][$i]);
69                     $tamanyo = $_FILES["ficheros"]["size"][$i];
70                     $tipo = $_FILES["ficheros"]["type"][$i];
71                     $usuario = $usuario;
72                     $cuota = $cuota;
73                     mysqli_stmt_execute($consulta1);
74                     mysqli_stmt_execute($consulta2);
75                 }
76                 else {
77                     $mensaje .= "Fichero \"".$_FILES["ficheros"]["name"][$i]."\n produce error desconocido 1.";
78                 }
79             }
80         else {
81             $mensaje .= "Fichero \"".$_FILES["ficheros"]["name"][$i]."\n excede el límite. ";
82         }
83         break;
84         case UPLOAD_ERR_NO_FILE:
85             $mensaje .= "Fichero inexistente.";
86             break;
87         case UPLOAD_ERR_INI_SIZE: case UPLOAD_ERR_FORM_SIZE:
88             $mensaje .= "Fichero \"".$_FILES["ficheros"]["name"][$i]."\n excede el límite. ";
89             break;
90         default:
91             $mensaje .= "Fichero \"".$_FILES["ficheros"]["name"][$i]."\n produce error desconocido 2.";
92             break;
93     }
94 }
95

```

Figura 9. Contenido del archivo “./www/paginaWeb/src/subirFicheros.php” (líneas 55-95).

Fuente: Elaboración propia.

Finalmente, cerramos ambas consultas, cerramos el canal y volvemos a la página del disco duro virtual con el mensaje correspondiente.

```

95
96     mysqli_stmt_close($consulta1);
97     mysqli_stmt_close($consulta2);
98     mysqli_close($canal);
99
100     header("Location: ../discoDuro.php?mensaje=".urlencode($mensaje))
101     ?>
102

```

Figura 9. Contenido del archivo “./www/paginaWeb/src/subirFicheros.php” (líneas 95-102).

Fuente: Elaboración propia.

## 2.1.6 DESCARGA DE FICHEROS

A la hora de descargar ficheros primero debemos recibir mediante el método POST el identificador del fichero a descargar, guardándolo en una variable local. También crearemos el canal con la base de datos.

```

1  <!-- Descargar los ficheros guardados en el disco duro -->
2
3  <?php
4      include "../../seguridad/ficheros/datosBD.php";
5
6  if (!isset($_POST["id"])){
7      header("Location: ../../discoDuro.php?mensaje=urlencode("ERROR DE SERVIDOR: No se ha podido recibir el ID del fichero a descargar.");");
8      exit;
9  }
10 $id = $_POST["id"];
11
12 $canal = @mysqli_connect(IP, USUARIO, CLAVE, BD);
13 if (!$canal){
14     echo "Ha ocurrido el error: " . mysqli_connect_errno() . " " . mysqli_connect_error() . "<br />";
15     exit;
16 }
17 mysqli_set_charset($canal, "utf8");
18

```

Figura 10. Contenido del archivo “./www/paginaWeb/src/descargarFicheros.php” (líneas 1-18).

Fuente: Elaboración propia.

Creamos una sentencia preparada para obtener las propiedades del fichero que buscamos y guardarlas en variables locales. Ejecutamos la consulta y guardamos en una variable el número de veces que se ha obtenido un resultado. Si por lo que fuera el fichero está duplicado o no existe (número de coincidencias distinto de 1), redireccionamos a la página del disco duro virtual. En el caso de que podamos continuar, modificamos la cabecera para indicar que hay un fichero adjunto y llamamos al “almacén” para poder adjuntar ese archivo. Finalmente cerramos la consulta y el canal.

```

18
19     $sql = "SELECT nombre, tamaño, tipo FROM ficheros WHERE id = ?";
20     $consulta = mysqli_prepare($canal, $sql);
21 if (!$consulta) {
22     echo "Ha ocurrido el error: " . mysqli_errno($canal) . " " . mysqli_error($canal) . "<br />";
23     exit;
24 }
25 mysqli_stmt_bind_param($consulta, "s", $id_);
26 mysqli_stmt_bind_result($consulta, $nombre_, $tamaño_, $tipo_);
27 $id_ = $id;
28 mysqli_stmt_execute($consulta);
29 mysqli_stmt_store_result($consulta);
30 $num = mysqli_stmt_num_rows($consulta);
31
32 if ($num != 1) {
33     header("Location: ../../discoDuro.php");
34     exit;
35 }
36 mysqli_stmt_fetch($consulta);
37
38 header("Content-disposition: attachment; filename = $nombre_");
39 header("Content-type: $tipo_");
40 readfile(__ALMACEN__.$id);
41
42 mysqli_stmt_close($consulta);
43 mysqli_close($canal);
44 ?>
45

```

Figura 11. Contenido del archivo “./www/paginaWeb/src/descargarFicheros.php” (líneas 18-45).

Fuente: Elaboración propia.

## 2.1.7 BORRADO DE FICHEROS

Para borrar los ficheros tenemos que seguir unos pasos iniciales similares a los de la descarga de ficheros. Debemos recibir mediante el método POST el identificador del fichero a borrar, guardándolo en una variable local. También crearemos el canal con la base de datos.

```

1 <!-- Borrar los ficheros guardados en el disco duro -->
2
3 <?php
4     include "../seguridad/ficheros/datosBD.php";
5     include "../seguridad/ficheros/funciones.php";
6     inicioSesion();
7
8     if (!isset($_POST["id"])){
9         header("Location: ../discoDuro.php?mensaje=".urlencode("ERROR DE SERVIDOR: No se ha podido recibir el ID del fichero a borrar."));
10        exit;
11    }
12    $id = $_POST["id"];
13
14    if (!isset($_SESSION["usuario"])) {
15        header("Location: ../discoDuro.php?mensaje=".urlencode("ERROR DE SERVIDOR: No se han podido recibir el usuario."));
16        exit;
17    }
18    $usuario = $_SESSION["usuario"];
19

```

Figura 12. Contenido del archivo “./www/paginaWeb/src/borrarFicheros.php” (líneas 1-19).

Fuente: Elaboración propia.

Preparamos dos consultas y las ejecutamos, la primera para guardarnos el tamaño del fichero que queremos borrar y la segunda para guardarnos la cuota actual del usuario que tiene la sesión activa.

```

19
20     $canal = @mysqli_connect(IP, USUARIO, CLAVE, BD);
21     if (!$canal){
22         echo "Ha ocurrido el error: ".mysqli_connect_errno()." ".mysqli_connect_error()."<br />";
23         exit;
24     }
25     mysqli_set_charset($canal, "utf8");
26
27     $sql = "SELECT tamaño FROM ficheros WHERE id = ?";
28     $consulta = mysqli_prepare($canal, $sql);
29     if (!$consulta) {
30         echo "Ha ocurrido el error: ".mysqli_errno($canal)." ".mysqli_error($canal)."<br />";
31         exit;
32     }
33     mysqli_stmt_bind_param($consulta, "s", $id);
34     $id_ = $id;
35     mysqli_stmt_execute($consulta);
36     mysqli_stmt_bind_result($consulta, $tamaño);
37     mysqli_stmt_fetch($consulta);
38     unset($consulta);
39
40     $sql = "SELECT cuota FROM usuarios WHERE usuario = ?";
41     $consulta = mysqli_prepare($canal, $sql);
42     if (!$consulta) {
43         echo "Ha ocurrido el error: ".mysqli_errno($canal)." ".mysqli_error($canal)."<br />";
44         exit;
45     }
46     mysqli_stmt_bind_param($consulta, "s", $usuario);
47     $usuario_ = $usuario;
48     mysqli_stmt_execute($consulta);
49     mysqli_stmt_bind_result($consulta, $cuota);
50     mysqli_stmt_fetch($consulta);
51     unset($consulta);
52

```

Figura 13. Contenido del archivo “./www/paginaWeb/src/borrarFicheros.php” (líneas 19-52).

Fuente: Elaboración propia.

Comprobamos que podemos borrar el archivo en el almacén. De ser así, preparamos otra consulta para borrar la fila de la tabla “ficheros” que se corresponda con el id del fichero que queremos borrar. Así eliminamos los datos del fichero tanto del almacén como de la base de datos. Una vez hecho esto, devolvemos a la cuota el valor del tamaño del fichero borrado y preparamos otra consulta para actualizar el campo de la cuota en la tabla “usuarios”. Cerramos la consulta, el canal y volvemos a la página del disco duro virtual.

```

52
53 ▼   if (unlink(__ALMACEN__.$id)) {
54       $sql = "DELETE FROM ficheros WHERE id = ?";
55       $consulta = mysqli_prepare($canal, $sql);
56 ▼   if (!$consulta) {
57       echo "Ha ocurrido el error: ".mysqli_errno($canal).". ".mysqli_error($canal)."<br />";
58       exit;
59   }
60   mysqli_stmt_bind_param($consulta, "s", $id_);
61   $id_ = $id;
62   mysqli_stmt_execute($consulta);
63   unset($consulta);
64
65   $cuota = $cuota + $tamanyo;
66
67   $sql = "UPDATE usuarios SET cuota = ? WHERE usuario = ?";
68   $consulta = mysqli_prepare($canal, $sql);
69 ▼   if (!$consulta) {
70       echo "Ha ocurrido el error: ".mysqli_errno($canal).". ".mysqli_error($canal)."<br />";
71       exit;
72   }
73   mysqli_stmt_bind_param($consulta, "is", $cuota_, $usuario_);
74   $cuota_ = $cuota;
75   $usuario_ = $usuario;
76   mysqli_stmt_execute($consulta);
77   unset($consulta);
78   }
79 ▼   else {
80       die("ERROR FATAL: No se ha podido eliminar el fichero.");
81   }
82
83   mysqli_stmt_close($consulta);
84   mysqli_close($canal);
85
86   // header("Location: ../../discoDuro.php?mensaje=".urlencode("Fichero borrado.));
87   header("Location: ../../discoDuro.php");
88   ?>
89

```

Figura 14. Contenido del archivo “./www/paginaWeb/src/borrarFicheros.php” (líneas 52-89).

Fuente: Elaboración propia.

## 2.1.8 PÁGINA DE LOGIN

La página de login “index.php” llama a un script PHP antes de cargar el código HTML. Este script comprueba que no haya una sesión activa en ese momento. De ser así, nos redireccionaría a la página del disco duro virtual.

```

1 <!-- Script previo a la utilización de la página index.php, si ya hay un usuario validado, nos manda a discoDuro.php -->
2
3 <?php
4     inicioSesion();
5     $usuario = "";
6     if (validado($usuario)) {
7         header("Location: ./discoDuro.php");
8         exit;
9     }
10 ?>
11

```

Figura 15. Contenido del archivo “./seguridad/ficheros/index-s.php”.

Fuente: Elaboración propia.

```

1 <!-- Página principal de login -->
2
3 <?php
4     include "../seguridad/ficheros/funciones.php";
5     include "../seguridad/ficheros/index-s.php";
6     $mensaje = "";
7     if (isset($_GET["mensaje"])) {
8         $mensaje = strip_tags(trim($_GET["mensaje"]));
9     }
10 ?>
11
12 <!DOCTYPE html>
13 <html>
14
15 <head>
16     <title>Iniciar sesión - Disco Duro Virtual en PHP</title>
17     <meta charset="UTF-8" />
18     <link rel="icon" href="./img/favicon.png" />
19     <link rel="stylesheet" type="text/css" href="./css/index.css" />
20 </head>
21
22 <body>
23     <header>
24         <a href="./index.php">
25             
26             
27         </a>
28     </header>
29     <section>
30         <article>
31             
32         </article>
33         <article>
34             <h3>Iniciar sesión</h3>
35             <p id="mensaje"><?=$mensaje?></p>
36             <form method="post" action="./src/validarSesion.php" autocomplete="off">
37                 <input type="text" placeholder="Usuario" name="usuario" maxlength="20" size="20" required="required" />
38                 <input type="password" placeholder="Contraseña" name="clave" maxlength="20" size="20" required="required" />
39                 <button type="submit">Iniciar sesión</button>
40             </form>
41         </article>
42     </section>
43     <footer>
44         <p>
45             "Disco duro virtual en PHP"<br />
46             DAW 2 - Desarrollo Web en Entorno Servidor<br />
47             I.E.S. Virgen del Espino (Soria)<br />
48             Enero 2019
49         </p>
50     </footer>
51 </body>
52 </html>
53
54

```

Figura 16. Contenido del archivo “./www/paginaWeb/index.php”.

Fuente: Elaboración propia.

## 2.1.9 PÁGINA DEL DISCO DURO VIRTUAL

La página del disco duro virtual “discoDuro.php” llama a un script PHP antes de cargar el código HTML. Este script comprueba que haya un usuario validado en ese momento. De no ser así, destruiría todas las sesiones activas y redirigiría a la página de login.

```

1  <!-- Script previo a la utilización de la página discoDuro.php, si no se valida el usuario, nos echa fuera -->
2
3  <?php
4      inicioSesion();
5      $usuario = "";
6  ▼  if (!validado($usuario)) {
7          session_destroy();
8          unset($_SESSION);
9          header("Location: ./index.php?mensaje=".urlencode("Se ha cerrado la sesión."));
10         exit;
11     }
12  ?>
13

```

Figura 17. Contenido del archivo “./seguridad/ficheros/discoDuro-s.php”.

Fuente: Elaboración propia.

Esta página realiza una consulta para obtener los datos del usuario que tiene la sesión activa y utilizarlos para mostrarlos en diferentes secciones de la pantalla. También realiza otra consulta para mostrar los archivos de ese usuario en forma de tabla, si los hubiera.

```

1  <!-- Página del disco duro -->
2
3  <?php
4      include "../seguridad/ficheros/datosBD.php";
5      include "../seguridad/ficheros/funciones.php";
6      include "../seguridad/ficheros/discoDuro-s.php";
7
8      $mensaje = "";
9  ▼  if (isset($_GET["mensaje"])) {
10         $mensaje = strip_tags(trim($_GET["mensaje"]));
11     }
12
13     $usuario = $_SESSION["usuario"];
14
15     $canal = @mysqli_connect(IP, USUARIO, CLAVE, BD);
16  ▼  if (!$canal) {
17         echo "Ha ocurrido el error: " . mysqli_connect_errno() . " " . mysqli_connect_error() . "<br />";
18         exit;
19     }
20     mysqli_set_charset($canal, "utf8");
21
22     $sql = "SELECT nombre, cuota FROM usuarios WHERE usuario = ?";
23     $consulta = mysqli_prepare($canal, $sql);
24  ▼  if (!$consulta) {
25         echo "Ha ocurrido el error: " . mysqli_errno($canal) . " " . mysqli_error($canal) . "<br />";
26         exit;
27     }
28     mysqli_stmt_bind_param($consulta, "s", $usuario);
29     $usuario_ = $usuario;
30     mysqli_stmt_execute($consulta);
31     mysqli_stmt_bind_result($consulta, $nombreUsuario, $cuota);
32     mysqli_stmt_fetch($consulta);
33     unset($consulta);
34
35     // $formatoCuota = number_format(($cuota / pow(2, 10)), 0, ".", ",");
36  ?>
37

```

Figura 18. Contenido del archivo “./www/paginaWeb/discoDuro.php” (líneas 1-37).

Fuente: Elaboración propia.



```

37
38 <!DOCTYPE html>
39 <html>
40
41 <head>
42 <title>Archivos - Disco Duro Virtual en PHP</title>
43 <meta charset="UTF-8" />
44 <link rel="icon" href="/img/favicon.png" />
45 <link rel="stylesheet" type="text/css" href="/css/discoDuro.css" />
46 </head>
47
48 <body>
49 <header>
50 <h3>Bienvenido, <?=$nombreUsuario?>!</h3>
51 </header>
52 <aside>
53 <a href="/discoDuro.php">
54 
55 </a>
56 <table>
57 <tr>
58 <td>Usuario:</td>
59 <td><?=$usuario?></td>
60 </tr>
61 <tr>
62 <td>Cuota:</td>
63 <td><?=$cuota?> bytes</td>
64 </tr>
65 </table>
66 <form method="post" action="/src/cerrarSesion.php">
67 <button type="submit" id="btnCerrarSesion">✕ Cerrar sesión</button>
68 </form>
69 </aside>

```

Figura 19. Contenido del archivo “./www/paginaWeb/discoDuro.php” (líneas 37-69).

Fuente: Elaboración propia.

```

70 <section>
71 <?php
72 $sql = "SELECT id, nombre, tamaño FROM ficheros WHERE usuario = ? ORDER BY nombre";
73 $consulta = mysqli_prepare($canal, $sql);
74 if (!$consulta) {
75     echo "Ha ocurrido el error: ".mysqli_errno($canal). " ".mysqli_error($canal). "<br />";
76     exit;
77 }
78 mysqli_stmt_bind_param($consulta, "s", $usuario);
79 $usuario = $usuario;
80 mysqli_stmt_execute($consulta);
81 mysqli_stmt_bind_result($consulta, $id, $nombre, $tamaño);
82 $cont = 0;
83 while (mysqli_stmt_fetch($consulta)) {
84     if ($cont == 0) {
85         echo "<table cellpadding='0'>";
86         <tr>
87             <th>Nombre</th>
88             <th>Tamaño</th>
89         </tr>";
90     }
91     echo "<tr>";
92     <td style='width:200px'>$nombre</td>
93     <td style='width:200px'>$tamaño bytes</td>
94     <td style='width:60px'>
95         <form method='post' action='./src/descargarFicheros.php'>
96             <input type='hidden' name='id' value='$id' />
97             <input type='image' src='./img/descargar.png' />
98         </form>
99     </td>
100     <td style='width:60px'>
101         <form method='post' action='./src/borrarFicheros.php'>
102             <input type='hidden' name='id' value='$id' />
103             <input type='image' src='./img/borrar.png' />
104         </form>
105     </td>
106     </tr>";
107     $cont++;
108 }
109 if ($cont == 0) {
110     echo "<table>";
111     <tr>
112         <th>No hay ficheros.</th>
113     </tr>";
114 }
115 echo "</table>";
116 mysqli_stmt_close($consulta);
117 unset($consulta);
118 ?>

```

Figura 20. Contenido del archivo “./www/paginaWeb/discoDuro.php” (líneas 70-118).

Fuente: Elaboración propia.

```

119 <p id="mensaje">&nbsp;<?=$mensaje?></p>
120 <form method="post" action="./src/subirFicheros.php" enctype="multipart/form-data">
121 <table cellpadding="0">
122 <tr>
123 <th>Subida de ficheros</th>
124 </tr>
125 <tr>
126 <td>Fichero(s):<input type="hidden" name="MAX_FILE_SIZE" value="<?=$cuota?>" /></td>
127 <td><input type="file" name="ficheros[]" multiple="multiple" /></td>
128 </tr>
129 <tr>
130 <td colspan="4" style="text-align:center;"><input type="submit" id="btnSubirFicheros" value="e Subir" id="btnSubir" /></td>
131 </tr>
132 </table>
133 </form>
134 </section>
135 </body>
136 </html>
137
138

```

Figura 21. Contenido del archivo “./www/paginaWeb/discoDuro.php” (líneas 118-128).

Fuente: Elaboración propia.

### 2.1.10 OTROS SCRIPTS

El siguiente script es un código previo a la llamada de “validarSesion-s.php”. Importa los archivos necesarios, así como redirige a la página del disco duro virtual una vez terminado.

```

1 <!-- Validar el usuario, fuera en la carpeta de seguridad -->
2
3 <?php
4 include "../../seguridad/ficheros/datosBD.php";
5 include "../../seguridad/ficheros/funciones.php";
6 include "../../seguridad/ficheros/validarSesion-s.php";
7 header("Location: ../../discoDuro.php");
8 ?>
9

```

Figura 22. Contenido del archivo “./seguridad/ficheros/validarSesion-s.php”.

Fuente: Elaboración propia.

## 3. MANUAL DE USUARIO

### 3.1. FUNCIONAMIENTO

Hay dos páginas web. En la primera se encuentra el login para la autenticación del usuario, habiendo dos campos (usuario y contraseña) y un botón para iniciar sesión. En la segunda se encuentra el disco duro virtual, con una barra lateral con los datos básicos del usuario, así como un botón para cerrar la sesión y volver a la página de login. También tiene una tabla con los ficheros pertenecientes a ese usuario, con la posibilidad de subir, descargar o borrar ficheros.

### **3.1.1 HACER LOGIN**

Para autenticarse en la aplicación es necesario estar registrado previamente y contar con un nombre de usuario y contraseña. Se deben introducir dichos datos en sus respectivos campos y pulsar el botón de iniciar sesión. En el caso de equivocación, se indicará esto y habrá la posibilidad de volver a introducir los datos. Una vez introducidos los datos correctamente, se mostrará la página del disco duro virtual.

### **3.1.2 SUBIR UN FICHERO**

Para subir un fichero al disco duro virtual, hay que seleccionar los archivos deseados mediante el botón designado para ello, que abrirá una ventana del explorador para poder hacerlo. Una vez seleccionado todo, se debe pulsar el botón de “Subir”. En el caso de que uno de los archivos supere la cuota permitida, se cancelará la subida y se avisará de ello.

### **3.1.3 DESCARGAR UN FICHERO**

Para descargar un fichero previamente subido al disco duro virtual, hay que pulsar el botón azul situado a la derecha del nombre de dicho archivo. Esta acción abrirá una ventana de diálogo para proceder a la descarga del archivo.

### **3.1.4 BORRAR UN FICHERO**

Para borrar un fichero previamente subido al disco duro virtual, hay que pulsar el botón rojo situado a la derecha del nombre de dicho archivo. Esta acción debe realizarse con cautela, pues no requiere de una segunda confirmación ni es reversible.

### 3.1.5 HACER LOGOUT

Para desautenticarse en la aplicación se debe pulsar el botón rojo con el texto “Cerrar sesión”.

Una vez hecho esto, se mostrará la página login.

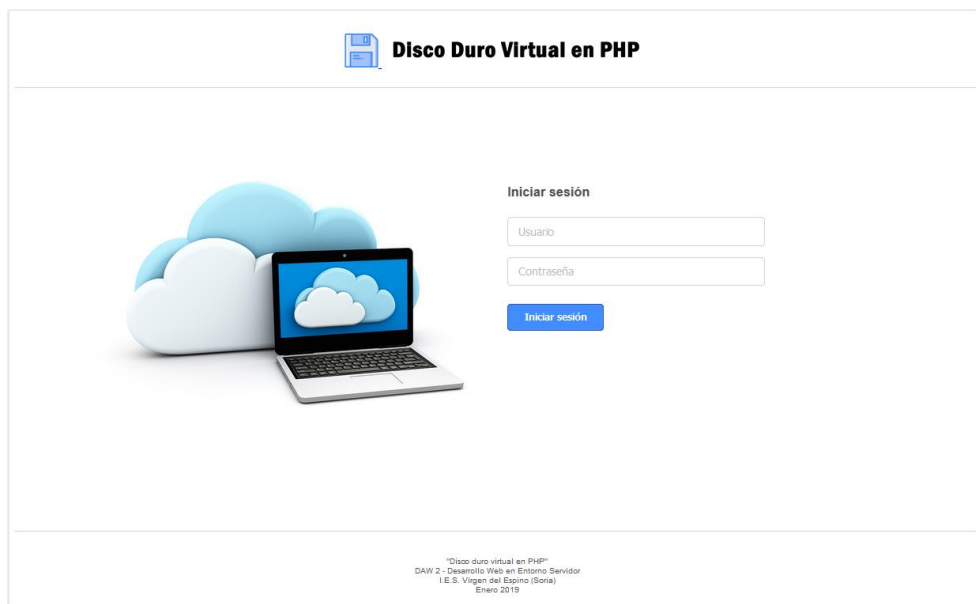


Figura 23. Vista de la página de login.

Fuente: Elaboración propia.

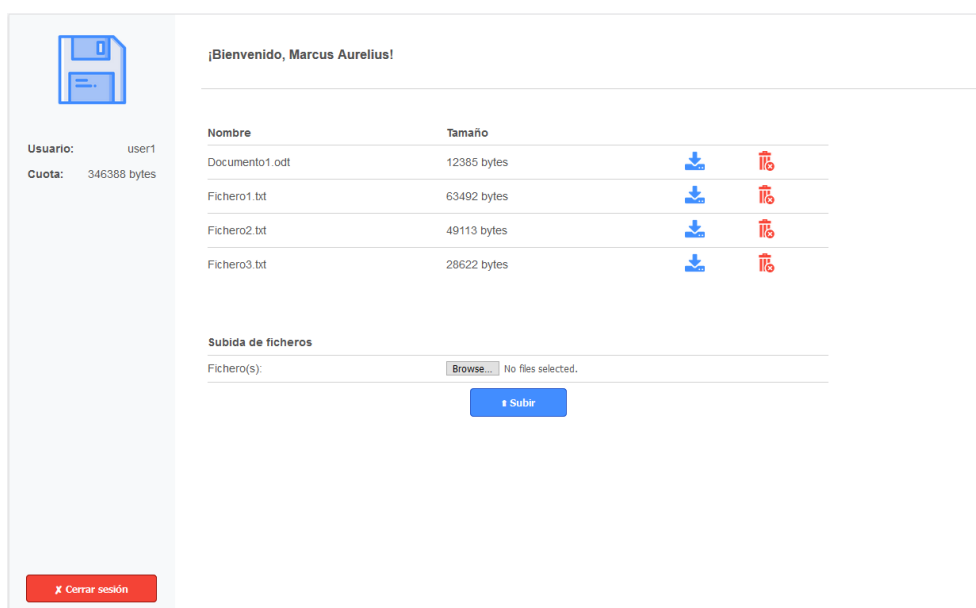


Figura 24. Vista de la página del disco duro virtual.

Fuente: Elaboración propia.

## BIBLIOGRAFÍA

### *Referencias bibliográficas:*

Homs Gimeno, J. (2018) Disco duro virtual en PHP. Material del módulo de Desarrollo de aplicaciones web en entorno servidor.

### *Direcciones y páginas web consultadas:*

<http://www.php.net/manual/en/>

<https://www.w3schools.com/>

<https://www.stackoverflow.com/>