



ShareMyMind

Tecnologías WEB

Luis Abdul Chávez Bustamante - 256910

25/11/2013

Introducción	3
Justificación	3
Descripción del proyecto	3
Montar la aplicación de manera externa	3
Herramientas de desarrollo	4
Interfaz del proyecto	5
Página de bienvenida	5
Registro	6
Autenticación	7
Pizarra	8
Configurar color de pensamiento	9
Votar por un pensamiento	10
Editar perfil	11
Validaciones	12
Página principal	13
Pizarra de usuario	15
Estructura de la base de datos	16
SQL	17
Estructura	17
Consultas	19
Estructura del proyecto	24
config.php	25
boot.php	27
stop.php	31
arrays.php (libs)	33
auth.php (libs)	34
database.php (libs)	41
redirect.php (libs)	57
smm.js (public/js)	58
board.php (public)	60

header.php (public)	70
index.php (public)	71
like.php (public)	75
login.php (public)	77
logout.php (public)	82
profile.php (public)	83
register.php (public)	97
share.php (public)	112
sidebar.php (public)	113
welcome.php (public)	114
<i>Glosario de funciones</i>	<i>117</i>

Introducción

ShareMyMind es un servicio para el intercambio de ideas de manera pública, cada usuario tiene su pizarra de ideas, El top de las mejores ideas se muestran en la página principal enlazando a la pizarra del autor.

Justificación

Para este proyecto fue conveniente utilizar una arquitectura cliente/servidor porque es necesario mantener un control centralizado sobre el contenido creado por los usuarios, de esta manera se puede acceder desde cualquier lugar sin necesidad de programas externos, solo mediante el uso del navegador web.

Descripción del proyecto

ShareMyMind permite compartir tus pensamientos de manera pública, crear nuevos pensamientos desde la pizarra, así como compartirla mediante la dirección única que enlaza a la pizarra de cada usuario, los usuarios puede votar una sola vez por cada pensamiento, los pensamientos más votados se muestran en la página principal.

La aplicación permite el registro de nuevos usuarios así como el acceso a la aplicación por medio de sus credenciales.

Los perfiles de usuario pueden ser editados desde la aplicación y el color personal de tus pensamientos puede ser cambiado desde la pizarra.

Para el uso de la imagen personal del usuario (avatar) se utiliza el servicio que proporciona **Gravatar**, el servicio utiliza el correo registrado en el perfil del usuario para mostrar la imagen de perfil correspondiente.

ShareMyMind no permite agregar amigos o establecer el nivel de privacidad de cada usuario, tampoco permite subir imágenes o eliminar pensamientos.

Montar la aplicación de manera externa

Para montar la aplicación de manera externa se debe de activar el servicio de apache de manera que puede ser accedido desde fuera de la red local, además de abrir los puertos correspondientes tanto en el firewall como en el router, por lo general el puerto a abrir es el puerto 80.

Herramientas de desarrollo

ShareMyMind está construido con las siguientes herramientas:

- MYSQL (<http://www.mysql.com/>)
- PHP (<http://php.net/>)
- HTML (HTML5) (<http://www.w3schools.com/html/>)
- CSS (Bootstrap) (<http://getbootstrap.com/>)
- JavaScript (jQuery) (<http://jquery.com/>)
- Gravatar (Avatar) (<http://es.gravatar.com/>)

Interfaz del proyecto

Para realizar la interfaz de la aplicación se utilizó bootstrap y jQuery.

Página de bienvenida

Cuando el usuario no está autenticado se muestra la página de bienvenida con los enlaces para registrar un nuevo usuario o autenticarse, en caso de que solo se quiera acceder a la página principal hay que acceder a dicho enlace.

ShareMyMind ¡Comparte tus ideas!

Ingresar

Registrarse

¡Bienvenido a ShareMyMind!

ShareMyMind es el lugar donde puedes compartir tus ideas al mundo sin limitaciones.

Ingresar

Registrarse

[Ingresar a la pagina principal](#)

Registro

Sección para el registro de nuevos usuarios, los campos requeridos son los siguientes:

- **Nombre** (Nombre del usuario por ejemplo: Luis)
- **Apellido** (Apellido del usuario por ejemplo: Chávez)
- **Fecha de nacimiento** (Por ejemplo: 22/01/1992)
- **Correo** (Correo del usuario, debe de ser único, se puede cambiar posteriormente)
- **Nombre de usuario** (Nombre de usuario, utilizado para autenticarse, debe de ser único)
- **Contraseña** (Contraseña del usuario)
- **Confirmar contraseña** (Confirmación de la contraseña)

ShareMyMind ¡Comparte tus ideas!

Ingresar Registrarse

[Inicio](#)

Registro

Nombre

Apellido

Fecha de nacimiento

dd/mm/aaaa

Correo

Nombre de usuario

Contraseña

Confirmar contraseña

Registrar

Cancelar

Autenticación

Sección de autenticación, para autenticar al usuario se necesitan de los siguientes datos:

- **Nombre de usuario** (Es el nombre de usuario no el nombre del usuario)
- **Contraseña**

ShareMyMind ¡Comparte tus ideas!

Ingresar

Registrarse

[Inicio](#)

Login

Usuario

Contraseña

☐ Recordar

Ingresar

Pizarra

Pizarra personal del usuario, en esta sección se pueden agregar pensamientos así como cambiar el color de los pensamientos.

ShareMyMind ¡Comparte tus ideas!



leviathan

Salir

[Inicio](#)

[Mi Perfil](#)

[Mi Pizarra](#)

Color



Actualizar



leviathan Fecha: 2013-11-21 21:01:54

Esto solo es para comprobar el funcionamiento.



De 20 a 120
caracteres

Enviar

Configurar color de pensamiento

Los pensamientos pueden tener mínimo 1 carácter y máximo 120, si se ingresan más de 120 caracteres solo se tomaran en cuenta los primeros 120.

Para cambiar el color de los pensamientos, hay que seleccionar el color y luego actualizar.

El color de los pensamientos se visualizara también en la página principal.

ShareMyMind ¡Comparte tus ideas!



leviathan

Salir

Inicio

Mi Perfil

Mi Pizarra

Color



Actualizar



leviathan Fecha: 2013-11-21 21:01:54

Esto solo es para comprobar el funcionamiento.



0

De 20 a 120
caracteres

Enviar

Votar por un pensamiento

Para votar por un pensamiento se debe de seleccionar el corazón que acompaña a cada uno de los pensamientos.

Si el usuario ya voto ese pensamiento el corazón se rellenara, si el usuario vuelve a votar por un pensamiento ya votado se eliminara el voto y el corazón pasara a estar vacío.

A un lado se muestra el contador de votos, mientras más votos tiene el pensamiento mayor posibilidad tiene de aparecer en la página principal.

ShareMyMind ¡Comparte tus ideas!



leviathan

Salir

[Inicio](#)

[Mi Perfil](#)

[Mi Pizarra](#)

Color



Actualizar



leviathan Fecha: 2013-11-21 21:01:54

Esto solo es para comprobar el funcionamiento.



De 20 a 120
caracteres

Enviar

Editar perfil

En esta sección el usuario puede cambiar su información personal y su contraseña.

Si el usuario cambia su correo la imagen de perfil también cambiara ya que se utiliza el servicio de **Gravatar** para el uso de imágenes de perfil.

ShareMyMind ¡Comparte tus ideas!



leviathan

Salir

[Inicio](#)

[Mi Perfil](#)

[Mi Pizarra](#)

Editar Perfil: leviathan



Nombre

Apellido

Fecha de nacimiento  

Correo

Nueva contraseña

Confirmar contraseña

Contraseña actual

Guardar

Cancelar

Validaciones

Las validaciones de formulario se muestran de la siguiente manera:

ShareMyMind ¡Comparte tus ideas!



leviathan

Salir

[Inicio](#)

[Mi Perfil](#)

[Mi Pizarra](#)

Editar Perfil: leviathan



Nombre

Luis

Apellido

Chavez

Fecha de
nacimiento

05/11/2013



Correo

Frost_Leviathan@hotmail.com

Nueva
contraseña

Confirmar
contraseña

Contraseña
actual

La contraseña no es valida..

Guardar

Cancelar



leviathan

Salir

[Inicio](#)

[Mi Perfil](#)

[Mi Pizarra](#)

Editar Perfil: leviathan



Nombre

Luis

Apellido

Chavez

Fecha de nacimiento

05/11/2013



Correo

Frost_Leviathan@hotmail.com

Nueva contraseña

Las contraseñas no coinciden.

Confirmar contraseña

Contraseña actual

Guardar

Cancelar



leviathan

Salir

[Inicio](#)

[Mi Perfil](#)

[Mi Pizarra](#)

Editar Perfil: leviathan



Nombre

Luis

Apellido

Chavez

Fecha de nacimiento

05/11/2013



Correo

Frost_Leviathan@hotmail.com

El correo ya esta en uso.

Nueva contraseña

Confirmar contraseña

Contraseña actual

Guardar

Cancelar

Página principal

La página principal es donde se muestra el top 20 de pensamientos de los usuarios, los pensamientos se ordenan de forma descendente de tal forma que los pensamientos con más votos se mostraran primero.

Cada pensamiento tiene un enlace a la pizarra del autor.

ShareMyMind ¡Comparte tus ideas!



luischavez

Salir

Inicio

Mi Perfil

Mi Pizarra



leviathan Fecha: 2013-11-21 21:01:54
Esto solo es para comprobar el funcionamiento.

♥ 2



luischavez Fecha: 2013-11-21 21:07:48
Comentario.

♥ 1

Pizarra de usuario

Sección que se mostrara a los usuarios que accedan a la pizarra de otro usuario.

ShareMyMind ¡Comparte tus ideas!



luischavez

Salir

Inicio

Mi Perfil

Mi Pizarra



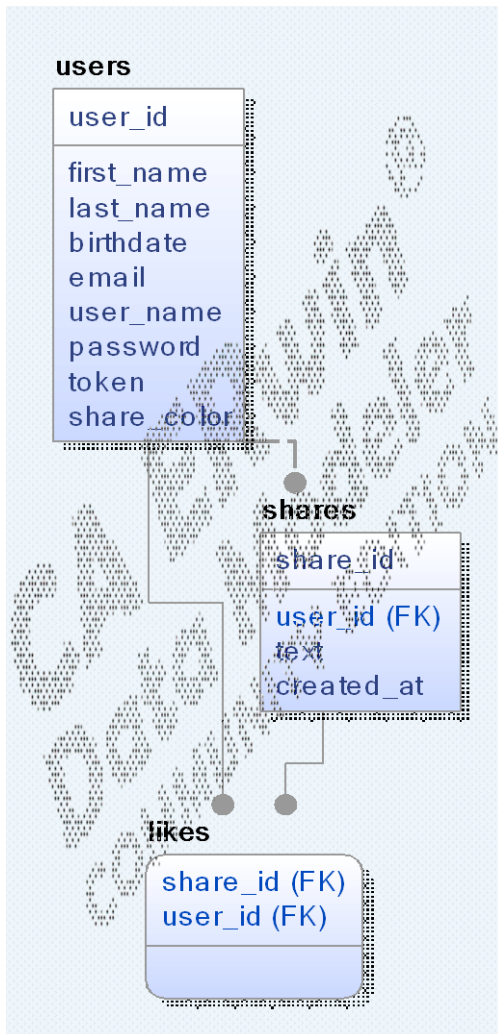
leviathan Fecha: 2013-11-21 21:01:54

Esto solo es para comprobar el funcionamiento.

♥ 2

Estructura de la base de datos

Para desarrollar el modelo de la base de datos se utilizó la herramienta Erwin Data Modeler.



SQL

Estructura

```
1 CREATE DATABASE IF NOT EXISTS smm DEFAULT CHARACTER SET utf8 COLLATE
  utf8_unicode_ci;
2 USE smm;
3
4 CREATE TABLE users
5 (
6     user_id          INTEGER NOT NULL,
7     user_name        VARCHAR(10) NOT NULL,
8     password         TEXT NOT NULL,
9     first_name       VARCHAR(20) NOT NULL,
10    last_name        VARCHAR(20) NOT NULL,
11    email            VARCHAR(50) NOT NULL,
12    token            TEXT NULL,
13    birthdate        DATE NOT NULL,
14    share_color       VARCHAR(20) NULL
15 );
16
17 ALTER TABLE users ADD PRIMARY KEY (user_id);
18 ALTER TABLE users CHANGE COLUMN user_id user_id INTEGER NOT NULL
  AUTO_INCREMENT;
19 ALTER TABLE users ADD UNIQUE (user_name);
20 ALTER TABLE users ADD UNIQUE (email);
21
22 CREATE TABLE shares
23 (
```

```
24     share_id          INTEGER NOT NULL,
25     text              VARCHAR(120) NOT NULL,
26     created_at        TIMESTAMP NOT NULL,
27     user_id           INTEGER NOT NULL
28 );
29
30 ALTER TABLE shares ADD PRIMARY KEY (share_id);
31 ALTER TABLE shares CHANGE COLUMN share_id share_id INTEGER NOT NULL
    AUTO_INCREMENT;
32
33 CREATE TABLE likes
34 (
35     share_id          INTEGER NOT NULL,
36     user_id           INTEGER NOT NULL
37 );
38
39 ALTER TABLE likes ADD PRIMARY KEY (share_id,user_id);
40
41 ALTER TABLE shares ADD FOREIGN KEY R_1 (user_id) REFERENCES users
    (user_id);
42
43 ALTER TABLE likes ADD FOREIGN KEY R_2 (share_id) REFERENCES shares
    (share_id);
44
45 ALTER TABLE likes ADD FOREIGN KEY R_3 (user_id) REFERENCES users
    (user_id);
```

Consultas

```
1 ----- Usuarios -----
2 -- Obtener el usuario a partir de su id.
3 SELECT *
4 FROM users
5 WHERE user_id = ?
6
7 -- Obtener el usuario a partir de su nombre de usuario.
8 SELECT *
9 FROM users
10 WHERE user_name = ?
11
12 -- Obtener el usuario a partir de su nombre de su email.
13 SELECT *
14 FROM users
15 WHERE email = ?
16
17 -- Actualizar el token de sesión del usuario.
18 UPDATE users
19 SET token = ?
20 WHERE user_id = ?
21
22 -- Crea un nuevo usuario.
23 INSERT INTO users
24     (first_name, last_name, birthdate, email, user_name, password)
25 VALUES (?, ?, ?, ?, ?, ?, ?)
```

```
26
27 -- Actualiza un usuario.
28 UPDATE users
29 SET
30     first_name = ?,
31     last_name = ?,
32     birthdate = ?,
33     email = ?,
34     password = ?
35 WHERE user_id = ?
36
37 -- Actualiza el color de los shares del usuario.
38 UPDATE users
39 SET
40     share_color = ?
41 WHERE user_id = ?
42 ----- Shares -----
43 -- Obtener el TOP de shares.
44 SELECT share_id
45 FROM likes
46 GROUP BY share_id
47 ORDER BY COUNT(share_id) DESC
48 LIMIT ?
49
50 -- Obtener un share.
51 SELECT *
```

```
52 FROM shares
53 WHERE share_id = ?
54
55 -- Obtener los ultimos shares.
56 SELECT *
57 FROM shares
58 ORDER BY created_at DESC
59 LIMIT ?
60
61 -- Obtener los shares paginados del usuario.
62 SELECT *
63 FROM shares
64 WHERE user_id = ?
65 ORDER BY created_at DESC
66 LIMIT ?, ?
67
68 -- Obtener el numero de shares del usuario.
69 SELECT COUNT(*) share_count
70 FROM shares
71 WHERE user_id = ?
72
73 -- Crea un nuevo share de parte de un usuario.
74 INSERT INTO shares
75     (user_id, text, created_at)
76 VALUES (?, ?, ?)
77
```

```
78 -- Elimina un share.
79 DELETE
80 FROM shares
81 WHERE share_id = ?
82 ----- Likes -----
83 -- Obtener el numero de likes de un share.
84 SELECT COUNT(share_id) AS likes
85 FROM likes
86 WHERE share_id = ?
87 GROUP BY share_id
88
89 -- Verifica si un usuario le dio like a un share.
90 SELECT 1 AS liked
91 FROM likes
92 WHERE share_id = ?
93     AND user_id = ?
94
95 -- Registra un nuevo like a un share de parte de un usuario.
96 INSERT INTO likes
97     (share_id, user_id)
98 VALUES (?, ?)
99
100 -- Elimina un like de un share de parte de un usuario.
101 DELETE
102 FROM likes
103 WHERE share_id = ?
```

104

AND user_id = ?

Estructura del proyecto

- ShareMyMind (**Proyecto**)
 - sql (**Scripts sql**)
 - ShareMyMind.sql (**Estructura de la base de datos**)
 - ShareMyMind_consultas.sql (**Lista de consultas que realiza la aplicación**)
 - src (**Código fuente**)
 - boot.php (**Inicia la aplicación**)
 - stop.php (**Detiene la aplicación, imprime mensajes de depuración**)
 - config.php (**Configuración de la aplicación**)
 - libs (**Librerías**)
 - arrays.php (**Librería para el manejo de arreglos**)
 - auth.php (**Librería para el manejo de la autenticación**)
 - database.php (**Librería para el manejo de la base de datos**)
 - redirect.php (**Librería para el manejo de redirecciones**)
 - public (**html, css, js**)
 - css (**Código css**)
 - fonts (**Fuentes**)
 - js (**Código javascript**)
 - header.php (**Cabecera de la aplicación**)
 - sidebar.php (**Barra lateral de la aplicación**)
 - like.php (**Registra o elimina un like**)
 - share.php (**Registra un nuevo share**)
 - index.php (**Muestra el TOP 20 de shares**)
 - board.php (**Pizarra del usuario**)
 - login.php (**Formulario de ingreso**)
 - logout.php (**Elimina la sesión**)
 - profile.php (**Muestra y actualiza el perfil del usuario**)
 - register.php (**Formulario de registro**)
 - welcome.php (**Página de bienvenida**)

config.php

```
1 <?php
2
3 /**
4  * Configuracion de la aplicacion.
5  */
6 $SMM = array(
7     // Modo de depuracion.
8     'debug' => FALSE,
9     // Titulo de la aplicacion.
10    'title' => 'ShareMyMind',
11    // Slogan de la aplicacion.
12    'slogan' => '¡Comparte tus ideas!',
13    // Lenguaje.
14    'lang' => 'es',
15    // Codificacion de caracteres.
16    'charset' => 'UTF8',
17    // Estilos css.
18    'styles' => array(
19        'css/bootstrap.min.css',
20    ),
21    // Javascript
22    'scripts' => array(
23        'js/jquery.min.js',
24        'js/bootstrap.min.js',
25        'js/holder.js',
```

```
26         'js/smm.js',
27     ),
28     // Codificación de la base de datos.
29     'database' => array(
30         // Host
31         'host' => 'localhost',
32         // Usuario
33         'user' => 'root',
34         // Password
35         'password' => '',
36         // Base de datos
37         'database' => 'smm',
38     ),
39     // Ruta completa a la carpeta raíz de la aplicación.
40     'app_path' => realpath(dirname(__FILE__)),
41     // Módulos necesarios para el funcionamiento de la
    aplicación.
42     // Se deben de agregar en orden de dependencia.
43     'auto_load' => array(
44         '/libs/database.php',
45         '/libs/auth.php',
46         '/libs/redirect.php',
47         '/libs/arrays.php',
48     ),
49 );
```

boot.php

```
1 <?php
2     /**
3      * Iniciamos la sesion.
4      */
5     session_start();
6
7     /**
8      * Incluimos el archivo de configuracion.
9      */
10    require_once '../config.php';
11
12    /**
13     * Cargamos todos los modulos de la aplicacion.
14     */
15    foreach ($SMM['auto_load'] as $module) {
16        require_once $SMM['app_path'] . $module;
17    }
18
19    /**
20     * Conectamos la base de datos.
21     */
22    $database->connect();
23
24    /**
25     * Verifica si existe la conexion con la base de datos.
```

```
26      * Si no existe termina la ejecucion.
27      */
28      if (!$database->isConnected()) {
29          die('No se puede realizar la conexion con la base de datos');
30      }
31
32      /**
33      * Inicia el sistema de sesion.
34      */
35      $auth->init();
36
37      /**
38      * Si esta en modo debug creamos las variables necesarias.
39      * Ademas iniciamos sesion en modo debug,
40      * en modo debug no es necesario establecer el password del
41      usuario.
42      */
43      if ($SMM['debug']) {
44
45          $GLOBALS['query_log'] = array();
46
47          $auth->login('luis', '');
48      }
49
50      /**
51      * Contamos el numero de visita.
52      */
```

```
51     $_SESSION['visit'] = isset($_SESSION['visit']) ?
$_SESSION['visit'] + 1 : 1;
52
53     /**
54     * Si es la primera visita se redirecciona a la pagina de
bienvenida.
55     */
56     if (1 == $_SESSION['visit']) {
57         $redirect->to('welcome.php');
58     }
59
60     /**
61     * Nombre del script que solicito el usuario.
62     */
63     $script = basename($_SERVER['SCRIPT_FILENAME']);
64
65     /**
66     * Si solicita el perfil, la pizarra o el logout y no esta
autenticado,
67     * se redirecciona a la pagina de bienvenida.
68     */
69     if ('profile.php' === $script || 'board.php' === $script ||
'logout.php' === $script) {
70         if (!$auth->isLogged()) {
71             $redirect->to('welcome.php');
72         }
73     }
```

```
74
75  /**
76   * Si solicita la pagina de bienvenida o login
77   * y esta autenticado, se redirecciona a la pagina principal.
78   */
79   if ('welcome.php' === $script || 'login.php' === $script) {
80       if ($auth->isLoggedIn()) {
81           $redirect->to('index.php');
82       }
83   }
```

stop.php

```
1 <?php
2
3 /**
4  * Si esta en modo debug mostramos la informacion de depuracion.
5  */
6  if ($SMM['debug']) {
7      echo '<div class="container">';
8      echo '<div class="row col-xs-6 col-xs-offset-3">';
9      echo '<div class="panel panel-info">';
10     echo '<div class="panel-heading">';
11     echo '<h3 class="panel-title">DEBUG</h3>';
12     echo '</div>';
13     echo '<div class="panel-body">';
14     foreach ($GLOBALS['query_log'] as $log) {
15         echo "<div class=\"alert alert-info\"><p>SQL:
16         <strong>{$log['sql']}</strong></p><p>VARS: <strong>{" . implode(' ',
17         $log['arguments']) . '}</strong></p></div>";
18     }
19     echo '</div>';
20     echo '</div>';
21     echo '</div>';
22     echo '</div>';
23 }
24
25 /**
26  * Cerramos la conexion con la base de datos.
```



```
25      */
```

```
26      $database->close();
```

arrays.php (libs)

```
1 <?php
2
3     /**
4      * Esta clase es necesaria ya que en algunas versiones de php
5      * no se puede acceder a un arreglo de la siguiente manera:
6      * $object->function()['key']
7      */
8
9     class Arrays {
10
11         /**
12          * Regresa el valor $key contenido en el arreglo $array
13          */
14         public function value($array, $key) {
15             return $array[$key];
16         }
17     }
18
19     /**
20      * Objeto con las funciones para manejar arreglos.
21      */
22
23     $arrays = new Arrays;
```

auth.php (libs)

```
1<?php
2
3    /**
4     * Clase con las funciones necesarias para manejar la
5     * autentificacion de los usuarios.
6     */
7
8    class Auth {
9
10        /**
11         * Funciones para manejar la base de datos.
12         */
13
14        protected $database;
15
16        function __construct() {
17
18            $this->database = $GLOBALS['database'];
19
20        }
21    }
```

```
17      /**
18      * Inicializa el sistema de sesiones.
19      * Si existen las cookies y son validas crea la sesion del
    usuario,
20      * si no son validas borra las cookies.
21      */
22      function init() {
23          if (isset($_SESSION['user'])) {
24              $_SESSION['user'] = $this->database-
    >getUserWithId($_SESSION['user']['user_id']);
25              return;
26          }
27
28          if (isset($_COOKIE['user_id']) &&
    isset($_COOKIE['user_token'])) {
29              $user = $this->database-
    >getUserWithUserName($_COOKIE['user_id']);
30
31              if (isset($user['token']) && $user['token'] ===
    $_COOKIE['user_token']) {
32                  $_SESSION['user'] = $user;
```

```
33         } else {
34             $this->logout(TRUE);
35         }
36     }
37 }
38
39 /**
40  * Verifica las credenciales del usuario y si son correctas
41  * realiza el login.
42  * Si remember es TRUE almacena las cookies necesarias.
43  * Si esta en modo de depuracion no guarda cookies.
44  */
45 function login($userName, $password, $remember = FALSE) {
46     if (isset($_SESSION['user'])) {
47         return TRUE;
48     }
```

```
49         $user = $this->database->getUserWithUserName($userName);

50

51         if (!$user) {

52             return FALSE;

53         }

54

55         if ($GLOBALS['SMM']['debug'] || $user['password'] ===
md5($password)) {

56             if ($remember) {

57                 $token =
substr(str_shuffle("0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLM
NOPQRSTUVWXYZ"), 0, 20);

58

59                 if ($this->database-
>setUserToken($user['user_id'], $token)) {

60                     if (!$GLOBALS['SMM']['debug']) {

61                         setcookie('user_id', $user['user_id'],
time() + (60 * 60 * 24 * 365));

62                         setcookie('user_token', $token, time() +
(60 * 60 * 24 * 365));

63                     }
```

```
64         }
65     }
66
67     $_SESSION['user'] = $this->database->
    >getUserWithUserName($userName);
68
69     return TRUE;
70 }
71
72     return FALSE;
73 }
74
75 /**
76  * Elimina la sesion del usuario y borra las cookies.
77  * Si $onlyCookies es TRUE solo se eliminan las cookies.
78  */
79     function logout($onlyCookies = FALSE) {
```

```

80         if (isset($_SESSION['user']) && !$onlyCookies) {

81             $user = $_SESSION['user'];

82             unset($_SESSION['user']);

83
84             $this->database->setUserToken($user['user_id'],
85             NULL);
86
87             setcookie('user_id', NULL, time() - (60 * 60 * 24 *
88             365));
89             setcookie('user_token', NULL, time() - (60 * 60 * 24 *
90             365));
91         }
92
93         /**
94          * Regresa el usuario que esta autenticado actualmente.
95          */
96
97         function getCurrentUser() {
98
99             return isset($_SESSION['user']) ? $_SESSION['user'] :
100             NULL;

```



```
96     }

97

98     /**

99     * Verifica si el usuario esta autentificado.

100    */

101    function isLoggedIn() {

102        return NULL != $this->getCurrentUser();

103    }

104 }

105

106 /**

107  * Objeto con las funciones de autentificacion.

108  */

109 $auth = new Auth;
```

database.php (libs)

```
1 <?php
2
3     /**
4      * Clase que contiene las funciones para manejar la base de
5      * datos.
6      */
7
8     class Database {
9
10         /**
11          * Conexion a la base de datos.
12          */
13
14         protected $connection = NULL;
15
16         /**
17          * Funcion para crear una nueva conexion.
18          */
19
20         function connect() {
21
22             $host = $GLOBALS['SMM']['database']['host'];
23             $user = $GLOBALS['SMM']['database']['user'];
24             $password = $GLOBALS['SMM']['database']['password'];
25             $database = $GLOBALS['SMM']['database']['database'];
26             $charset = $GLOBALS['SMM']['charset'];
27
28             $this->connection = mysql_connect($host, $user,
29 $password);
```

```
25         mysql_set_charset($charset);
26
27         if (!mysql_select_db($database, $this->connection)) {
28             $this->createDatabase();
29         }
30     }
31
32     /**
33     * Funcion para liberar la conexion.
34     */
35     function close() {
36         if (NULL != $this->connection) {
37             mysql_close($this->connection);
38         }
39     }
40
41     /**
42     * Verifica si existe una conexion valida.
43     */
44     function isConnected() {
45         return NULL != $this->connection;
46     }
47
48     /**
49     * Crea la estructura de la aplicacion en la base de datos.
50     */
```

```

51     function createDatabase() {
52         if ($this->isConnected()) {
53             $script = $GLOBALS['SMM']['app_path'] .
54                 './sql/ShareMyMind.sql';
55
56             $queries = explode(';', file_get_contents($script));
57
58             foreach ($queries as $query) {
59                 mysql_query($query);
60             }
61         }
62
63         /**
64          * Funcion para limpiar los argumentos de la consulta.
65          * Esto impide crear consultas inseguras.
66          */
67         function clean($arguments) {
68             for ($i = 0; $i < count($arguments); $i++) {
69                 if (is_string($arguments[$i])) {
70                     $arguments[$i] =
71                     mysql_escape_string($arguments[$i]);
72                 }
73             }
74
75             return $arguments;
76         }

```

```

76
77      /**
78      * Funcion para ejecutar una sentencia sql.
79      */
80      function executeQuery($sql, $arguments) {
81          $arguments = $this->clean($arguments);
82
83          if ($GLOBALS['SMM']['debug']) {
84              array_push($GLOBALS['query_log'], array('sql' =>
85                  $sql, 'arguments' => $arguments));
86          }
87
88          foreach ($arguments as $argument) {
89              if (NULL === $argument) {
90                  $sql = preg_replace('/\?/', 'NULL', $sql, 1);
91              } else if (is_numeric($argument)) {
92                  $sql = preg_replace('/\?/', "$argument", $sql,
93                      1);
94              } else {
95                  $sql = preg_replace('/\?/', "'$argument'", $sql,
96                      1);
97              }
98          }
99
100          $results = mysql_query($sql, $this->connection);
101
102          return $results;

```

```

100     }
101
102     /**
103      * Funcion para guardar los resultados en un arreglo y
liberar los recursos.
104      */
105     function getArray($results) {
106         $rows = array();
107
108         while ($row = mysql_fetch_assoc($results)) {
109             foreach ($row as $key => $value) {
110                 $rows[$key] = $value;
111             }
112         }
113
114         mysql_free_result($results);
115
116         return $rows;
117     }
118
119     /**
120      * Funcion para guardar los resultados en un arreglo
bidimensional y liberar los recursos.
121      */
122     function getArrays($results) {
123         $rows = array();
124

```

```

125         while ($row = mysql_fetch_assoc($results)) {
126             array_push($rows, $row);
127         }
128
129         mysql_free_result($results);
130
131         return $rows;
132     }
133
134     /**
135      * Ejecuta una consulta y devuelve un arreglo con la fila
136      obtenida.
137      */
138     function executeQueryAndGetArray($sql, $arguments) {
139
140         $results = $this->executeQuery($sql, $arguments);
141
142         if ($results) {
143             return $this->getArray($results);
144         }
145
146         return NULL;
147     }
148
149     /**
150      * Ejecuta una consulta y devuelve un arreglo bidimensional
151      con las filas obtenidas.
152      */

```

```

150     function executeQueryAndGetArrays($sql, $arguments) {
151         $results = $this->executeQuery($sql, $arguments);
152
153         if ($results) {
154             return $this->getArrays($results);
155         }
156
157         return array();
158     }
159
160     /**
161     * Retorna un arreglo con los usuarios,
162     * delimitado por $take y $page.
163     */
164     function getUsers($take = 10, $page = 1) {
165         $sql = "SELECT * FROM users LIMIT ?,?";
166
167         return $this->executeQueryAndGetArrays($sql, array($take
168 * ($page - 1), $take));
169     }
170
171     /**
172     * Retorna el usuario con el id = $user_id
173     */
174     function getUserWithId($userId) {
175         $sql = "SELECT * FROM users WHERE user_id = ?";

```



```
175
176         return $this->executeQueryAndGetArray($sql,
array($userId));
177     }
178
179     /**
180      * Retorna el usuario con el user_name = $userName
181      */
182     function getUserWithUserName($userName) {
183         $sql = "SELECT * FROM users WHERE user_name = ?";
184
185         return $this->executeQueryAndGetArray($sql,
array($userName));;
186     }
187
188     /**
189      * Retorna el usuario con el user_name = $userName
190      */
191     function getUserWithEmail($email) {
192         $sql = "SELECT * FROM users WHERE email = ?";
193
194         return $this->executeQueryAndGetArray($sql,
array($email));;
195     }
196
197     /**
198      * Establece el token de sesion del usuario.
```

```

199     */
200     function setUserToken($userId, $token) {
201         $sql = "UPDATE users SET token = ? WHERE user_id = ?";
202
203         return $this->executeQuery($sql, array($token,
204 $userId));
205     }
206
207     /**
208      * Obtiene un share.
209     */
210     function getShare($shareId) {
211
212         $sql = "SELECT * FROM shares WHERE share_id = ?";
213
214         return $this->executeQueryAndGetArray($sql,
215 array($shareId));
216     }
217
218     /**
219      * Obtiene el top de shares mas votados.
220      * Esta delimitado por $take.
221     */
222     function getTopShares($take = 20) {
223
224         $sql = "SELECT share_id FROM likes GROUP BY share_id
225 ORDER BY COUNT(share_id) DESC LIMIT ?";
226
227         $results = $this->executeQueryAndGetArrays($sql,

```

```

    array($take));
223
224         if (0 === count($results)) {
225             $sql = "SELECT * FROM shares ORDER BY created_at
DESC LIMIT ?";
226
227             $results = $this->executeQueryAndGetArrays($sql,
array($take));
228         } else {
229             foreach ($results as &$share) {
230                 $share = $this->getShare($share['share_id']);
231             }
232         }
233
234         return $results;
235     }
236
237     /**
238      * Obtiene los shares del usuario paginados.
239      */
240     function getUserShares($userId, $take = 10, $page = 1) {
241         $sql = "SELECT * FROM shares WHERE user_id = ? ORDER BY
created_at DESC LIMIT ?, ?";
242
243         return $this->executeQueryAndGetArrays($sql,
array($userId, $take * ($page - 1), $take));
244     }

```

```
245
246     /**
247     * Obtiene el numero de shares del usuario.
248     */
249     function getShareCount($userId) {
250         $sql = "SELECT COUNT(*) AS share_count FROM shares WHERE
251 user_id = ?";
252
253         return $this->executeQueryAndGetArray($sql,
254 array($userId));
255     }
256
257     /**
258     * Obtiene el numero de likes de un share.
259     */
260     function getLikeCount($shareId) {
261         $sql = "SELECT COUNT(share_id) AS likes FROM likes WHERE
262 share_id = ? GROUP BY share_id";
263
264         $result = $this->executeQueryAndGetArray($sql,
265 array($shareId));
266
267         return $result ? $result['likes'] : 0;
268     }
269
270     /**
271     * Verifica si un usuario le dio like a un share.
```

```

268         */
269         function isLikedBy($shareId, $userId) {
270             $sql = "SELECT 1 AS liked FROM likes WHERE share_id = ?
AND user_id = ?";
271
272             $result = $this->executeQueryAndGetArray($sql,
array($shareId, $userId));
273
274             return $result ? TRUE : FALSE;
275         }
276
277         /**
278          * Crea un like para el share de parte del usuario.
279          */
280         function like($shareId, $userId) {
281             $sql = "INSERT INTO likes (share_id, user_id) VALUES
(?, ?)";
282
283             return $this->executeQuery($sql, array($shareId,
$userId));
284         }
285
286         /**
287          * Elimina el like de un share de parte del usuario.
288          */
289         function unlike($shareId, $userId) {
290             $sql = "DELETE FROM likes WHERE share_id = ? AND user_id
= ?";

```

```
291
292         return $this->executeQuery($sql, array($shareId,
293         $userId));
294
295     }
296
297     /**
298     * Obtiene el avatar del usuario a partir del email.
299     * Utiliza el servicio de gravatar.
300     */
301
302     function getUserAvatar($userId) {
303         $user = $this->getUserWithId($userId);
304
305         if ($user) {
306             return "http://www.gravatar.com/avatar/" .
307             md5(strtolower(trim($user['email']))) . "?d=mm";
308         }
309
310         return "http://www.gravatar.com/avatar?d=mm";
311     }
312
313     /**
314     * Crea un nuevo share.
315     */
316
317     function share($userId, $text) {
318         $sql = "INSERT INTO shares (user_id, text, created_at)
319         VALUES (?, ?, ?)";
320
321     }
```

```

315         return $this->executeQuery($sql, array($userId,
htmlentities($text), date('Y-m-d H:i:s'))));
316     }
317
318     /**
319     * Actualiza la informacion del usuario.
320     */
321     function updateUser($user) {
322         $sql = "UPDATE users SET first_name = ?, last_name = ?,
birthdate = ?, email = ?, password = ? WHERE user_id = ?";
323
324         $arguments = array(
325             $user['first_name'],
326             $user['last_name'],
327             $user['birthdate'],
328             $user['email'],
329             $user['password'],
330             $user['user_id'],
331         );
332
333         return $this->executeQuery($sql, $arguments);
334     }
335
336     /**
337     * Registra un nuevo usuario.
338     */
339     function registerUser($user) {

```

```

340         $sql = "INSERT INTO users (first_name, last_name,
birthdate, email, user_name, password) VALUES (?, ?, ?, ?, ?, ?)";
341
342         $arguments = array(
343             $user['first_name'],
344             $user['last_name'],
345             $user['birthdate'],
346             $user['email'],
347             $user['user_name'],
348             $user['password'],
349         );
350
351         return $this->executeQuery($sql, $arguments);
352     }
353
354     /**
355      * Actualiza el color de los shares del usuario.
356      */
357     function setShareColor($userId, $color = NULL) {
358         $sql = "UPDATE users SET share_color = ? WHERE user_id
= ?";
359
360         return $this->executeQuery($sql, array($color,
$userId));
361     }
362 }
363

```



```
364     /**
365     * Objeto con las funciones necesarias para interactuar con la
    base de datos.
366     */
367     $database = new Database;
```

redirect.php (libs)

```
1 <?php
2
3 /**
4  * Clase necesaria para manejar redirecciones,
5  * cualquier cambio en esta clase cambia el comportamiento general
6  * de la aplicacion.
7  */
8
9  class Redirect {
10
11      function to($script) {
12          header("location: $script");
13          exit();
14      }
15  }
16
17 /**
18  * Objeto con las funciones necesarias para hacer redirecciones.
19  */
20
21 $redirect = new Redirect;
```

smm.js (public/js)

```
1 /**
2  * Cuando el documento este listo agregamos las funciones necesarias.
3  */
4  (function($) {
5      $("a.like").click(function() {
6          $like = $(this);
7          $.ajax({
8              type: "POST",
9              url: "like.php",
10             data: {
11                 user_id: $(this).data("user"),
12                 share_id: $(this).data("share")
13             }
14         }).done(function(liked) {
15             var likes =
16             parseInt($like.parent().children(".badge").text(), 10);
17
18             $like.parent().children(".badge").text(liked == 1 ? likes
19             + 1 : likes - 1);
20
21             if (liked == 1) {
22                 $like.attr("class", "glyphicon glyphicon-heart");
23             } else {
24                 $like.attr("class", "glyphicon glyphicon-heart-
25                 empty");
26             }
27         });
28     });
29 }
```

```
24     });
```

```
25 })(jQuery);
```

board.php (public)

```
1 <?php
2     /**
3      * Incluimos el archivo de arranque de la aplicacion.
4      * Es muy importante por que contiene las funciones
5      * necesarias para que trabaje correctamente la aplicacion.
6      */
7     require_once '../boot.php';
8 ?>
9 <?php
10    /**
11     * Si se pasa el nombre del usuario por $_GET
12     * se selecciona al usuario al que pertenece,
13     * en caso contrario se selecciona el usuario autenticado.
14     */
15    if (isset($_GET['user'])) {
16        $user = $database->getUserWithUserName($_GET['user']);
17    } else {
18        $user = $auth->getCurrentUser();
19    }
20
21    /**
22     * Si el usuario no existe redirecciona a la pagina principal.
23     */
24    if (NULL == $user) {
25        $redirect->to('index.php');
```

```
26     }
27
28     /**
29     * Nombre de usuario.
30     */
31     $userName = $user['user_name'];
32
33     /**
34     * Si el numero de pagina se pasa por $_GET
35     * se utiliza para paginar los resultados,
36     * en caso contrario la pagina es igual a 1.
37     */
38     $page = isset($_GET['page']) ? $_GET['page'] : 1;
39
40     /**
41     * Si el numero de pagina no es un numero
42     * se redirecciona a la pagina 1.
43     */
44     if (!is_numeric($page)) {
45         $redirect->to("board.php?user=$userName&page=1");
46     }
47
48     /**
49     * Si el numero de pagina es menor o igual a 0
50     * se redirecciona a la pagina 1.
51     */
52     if (0 >= $page) {
```

```
52         $redirect->to("board.php?user=$userName&page=1");
53     }
54
55     /**
56     * Numero de resultados por pagina.
57     */
58     $perPage = 10;
59
60     /**
61     * Numero de items a paginar.
62     */
63     $shareCount = $database->getShareCount($user['user_id']);
64     $shareCount = $shareCount['share_count'];
65
66     /**
67     * Total de paginas.
68     */
69     $pages = ceil($shareCount / $perPage);
70
71     /**
72     * Si la pagina no existe y existen registros.
73     * redirecciona a la ultima pagina.
74     */
75     if (0 < $shareCount && $pages < $page) {
76         $redirect->to("board.php?user=$userName&page=$pages");
77     }
```

```

78
79  /**
80  * Si se envia el color por POST,
81  * se actualiza el color del usuario.
82  */
83  if (isset($_POST['share_color'])) {
84      if ($database->setShareColor($arrays->value($auth-
>getCurrentUser(), 'user_id'), $_POST['share_color'])) {
85          $user['share_color'] = $_POST['share_color'];
86      }
87  }
88  ?>
89  <!doctype html>
90  <!-- Establecemos el lenguaje de la aplicacion -->
91  <html lang="<?php echo $SMM['lang']; ?>">
92  <head>
93      <!-- Establecemos la codificacion de caracteres -->
94      <meta charset="<?php echo $SMM['charset']; ?>">
95      <!-- No escalable -->
96      <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
97      <!-- Establecemos el titulo de la aplicacion -->
98      <title><?php echo $SMM['title']; ?></title>
99      <!-- Incluimos los estilos css -->
100     <?php foreach ($SMM['styles'] as $style): ?>
101         <link rel="stylesheet" href="<?php echo $style; ?>">
102     <?php endforeach; ?>

```



```

103 </head>
104 <body>
105     <!-- Contenedor principal -->
106     <div class="container">
107         <!-- Cabecera de la aplicacion -->
108         <div class="row">
109             <div class="col-xs-12">
110                 <?php include 'header.php'; ?>
111             </div>
112         </div>
113         <!-- Cuerpo de la aplicacion -->
114         <div class="row">
115             <!-- Menu de navegacion -->
116             <div class="col-xs-3">
117                 <?php include 'sidebar.php'; ?>
118             </div>
119             <!-- Contenido -->
120             <div class="col-xs-9">
121                 <?php if ($auth->isLoggedIn() && $user['user_id'] ==
$arrays->value($auth->getCurrentUser(), 'user_id')): ?>
122                     <!-- Formulario share color -->
123                     <form class="form-inline" action="board.php"
method="POST">
124                         <div class="form-group">
125                             <label class=""
for="share_color">Color</label>
126                             <input type="color" name="share_color"
value="<?php echo isset($user['share_color']) ?

```

```

126 $user['share_color'] : ''; ?>">
127         </div>
128         <input class="btn btn-xs btn-info"
type="submit" value="Actualizar">
129     </form>
130     <?php endif; ?>
131     <!-- Shares -->
132     <ul class="media-list">
133         <?php foreach ($database->
getUserShares($user['user_id'], $perPage, $page) as $share): ?>
134             <li class="media" <?php echo
isset($user['share_color']) ? 'style="background-color: ' .
$user['share_color'] . '";" : ''; ?>>
135                 <a class="pull-left" href="#">
136                     
137                 </a>
138                 <div class="media-body">
139                     <h4 class="media-heading">
140                         <?php echo $user['user_name']; ?>
141                         <small>Fecha: <?php echo
$share['created_at']; ?></small>
142                     </h4>
143                     <p><?php echo $share['text']; ?></p>
144                     <p>
145                         <?php if ($auth->isLoggedIn()): ?>
146                             <?php if ($database->
isLikedBy($share['share_id'], $arrays->value($auth->

```

```

147         <a class="like glyphicon glyphicon-heart" data-user="<?php echo $arrays->value($auth->getCurrentUser(), 'user_id'); ?>" data-share="<?php echo $share['share_id']; ?>" href="#"></a>

148         <?php else: ?>

149         <a class="like glyphicon glyphicon-heart-empty" data-user="<?php echo $arrays->value($auth->getCurrentUser(), 'user_id'); ?>" data-share="<?php echo $share['share_id']; ?>" href="#"></a>

150         <?php endif; ?>

151         <?php endif; ?>

152         <span class="badge"><?php echo $database->getLikeCount($share['share_id']); ?></span>

153         </p>

154     </div>

155 </li>

156 <?php endforeach; ?>

157 </ul>

158 <!-- Paginacion -->

159 <ul class="pagination">

160     <?php if (0 < $pages && $page != 1): ?>

161         <li><a href="?user=<?php echo $userName; ?>&page=1">&laquo;</a></li>

162     <?php endif; ?>

163     <?php if (1 < $pages): ?>

164         <?php for ($i = 1; $i < $pages + 1; $i++): ?>

165             <?php if ($i == $page): ?>

```

```

166         <li class="active">
167             <a href="?user=<?php echo
$userName; ?>&page=<?php echo $i; ?>">
168                 <?php echo $i; ?>
169             </a>
170         </li>
171     <?php else: ?>
172         <li>
173             <a href="?user=<?php echo
$userName; ?>&page=<?php echo $i; ?>">
174                 <?php echo $i; ?>
175             </a>
176         </li>
177     <?php endif; ?>
178 <?php endfor; ?>
179 <?php endif; ?>
180 <?php if (0 < $pages && $page != $pages): ?>
181     <li><a href="?user=<?php echo
$userName; ?>&page=<?php echo $pages; ?>">&raquo;</a></li>
182 <?php endif; ?>
183 </ul>
184 <!-- Formulario para agregar shares -->
185 <?php if ($auth->isLogged()): ?>
186     <?php if ($user['user_id'] == $arrays-
>value($auth->getCurrentUser(), 'user_id')): ?>
187         <div class="row">
188             <div class="col-xs-8 col-xs-offset-2">

```

```
189         <form class="form-inline"
role="form" action="share.php" method="POST">
190             <div class="form-group">
191                 <textarea class="form-
control" rows="3" placeholder="De 20 a 120 caracteres"
name="text"></textarea>
192             </div>
193             <button type="submit" class="btn
btn-primary">Enviar</button>
194         </form>
195     </div>
196 </div>
197 <?php endif; ?>
198 <?php endif; ?>
199 </div>
200 </div>
201 </div>
202
203 <!-- Incluimos el codigo javascript -->
204 <?php foreach ($SMM['scripts'] as $script): ?>
205     <script src="<?php echo $script; ?>"></script>
206 <?php endforeach; ?>
207
208 <?php
209     /**
210     * Incluimos el archivo para finalizar la ejecucion de la
aplicacion.
211     * Es muy importante para manejar la aplicacion en modo de
```

depuracion.

212 */

213 require_once '../stop.php';

214 ?>

215 </body>

216 </html>

header.php (public)

```
1 <div class="media page-header">
2     <h1><?php echo $SMM['title']; ?> <small><?php echo
    $SMM['slogan']; ?></small></h1>
3     <?php if ($auth->isLoggedIn()): ?>
4         <div class="pull-left">
5             
6         </div>
7         <div class="media-body">
8             <h2 class="media-heading">
9                 <?php echo $arrays->value($auth->getCurrentUser(),
    'user_name'); ?>
10            </h2>
11            <a class="btn btn-warning btn-xs"
    href="logout.php">Salir</a>
12        </div>
13    <?php else: ?>
14        <p class="pull-right">
15            <a class="btn btn-primary btn-sm"
    href="login.php">Ingresar</a>
16            <a class="btn btn-default btn-sm"
    href="register.php">Registrarse</a>
17        </p>
18    <?php endif; ?>
19 </div>
```

index.php (public)

```
1 <?php
2     /**
3      * Incluimos el archivo de arranque de la aplicacion.
4      * Es muy importante por que contiene las funciones
5      * necesarias para que trabaje correctamente la aplicacion.
6      */
7     require_once '../boot.php';
8 ?>
9 <!doctype html>
10 <!-- Establecemos el lenguaje de la aplicacion -->
11 <html lang="<?php echo $SMM['lang']; ?>">
12 <head>
13     <!-- Establecemos la codificacion de caracteres -->
14     <meta charset="<?php echo $SMM['charset']; ?>">
15     <!-- No escalable -->
16     <meta name="viewport" content="width=device-width, initial-
17 scale=1.0, maximum-scale=1.0, user-scalable=no">
18     <!-- Establecemos el titulo de la aplicacion -->
19     <title><?php echo $SMM['title']; ?></title>
20     <!-- Incluimos los estilos css -->
21     <?php foreach ($SMM['styles'] as $style): ?>
22         <link rel="stylesheet" href="<?php echo $style; ?>">
23     <?php endforeach; ?>
24 </head>
25 <body>
```



```

25     <!-- Contenedor principal -->
26     <div class="container">
27         <!-- Cabecera de la aplicacion -->
28         <div class="row">
29             <div class="col-xs-12">
30                 <?php include 'header.php'; ?>
31             </div>
32         </div>
33         <!-- Cuerpo de la aplicacion -->
34         <div class="row">
35             <!-- Menu de navegacion -->
36             <div class="col-xs-3">
37                 <?php include 'sidebar.php'; ?>
38             </div>
39             <!-- Contenido -->
40             <div class="col-xs-9">
41                 <!-- Shares -->
42                 <ul class="media-list">
43                     <?php foreach ($database->getTopShares() as
$share): ?>
44                         <?php $user = $database-
>getUserWithId($share['user_id']); ?>
45                         <li class="media" <?php echo
isset($user['share_color']) ? 'style="background-color: ' .
$user['share_color'] . '";" : ';" ?>>
46                             <a class="pull-left" href="#">
47                             
48             </a>
49             <div class="media-body">
50                 <h4 class="media-heading">
51                     <a href="board.php?user=<?php echo
$user['user_name']; ?>">
52                         <?php echo $user['user_name']; ?>
53                     </a>
54                     <small>Fecha: <?php echo
$share['created_at']; ?></small>
55                 </h4>
56                 <p><?php echo $share['text']; ?></p>
57                 <p>
58                     <?php if ($auth->isLogged()): ?>
59                         <?php if ($database-
>isLikedBy($share['share_id'], $arrays->value($auth-
>getCurrentUser(), 'user_id')): ?>
60                             <a class="like glyphicon
glyphicon-heart" data-user="<?php echo $arrays->value($auth-
>getCurrentUser(), 'user_id'); ?>" data-share="<?php echo
$share['share_id']; ?>" href="#"></a>
61                         <?php else: ?>
62                             <a class="like glyphicon
glyphicon-heart-empty" data-user="<?php echo $arrays->value($auth-
>getCurrentUser(), 'user_id'); ?>" data-share="<?php echo
$share['share_id']; ?>" href="#"></a>
63                         <?php endif; ?>
64                     <?php endif; ?>
65                     <span class="badge"><?php echo
$database->getLikeCount($share['share_id']); ?></span>

```

```

66         </p>
67     </div>
68 </li>
69 <?php endforeach; ?>
70 </ul>
71 </div>
72 </div>
73 </div>
74
75 <!-- Incluimos el codigo javascript -->
76 <?php foreach ($SMM['scripts'] as $script): ?>
77     <script src="<?php echo $script; ?>"></script>
78 <?php endforeach; ?>
79
80 <?php
81     /**
82      * Incluimos el archivo para finalizar la ejecucion de la
83      * aplicacion.
84      * Es muy importante para manejar la aplicacion en modo de
85      * depuracion.
86      */
87     require_once '../stop.php';
88     ?>
89 </body>
90 </html>

```

like.php (public)

```
1 <?php
2     /**
3      * Incluimos el archivo de arranque de la aplicacion.
4      * Es muy importante por que contiene las funciones
5      * necesarias para que trabaje correctamente la aplicacion.
6      */
7     require_once '../boot.php';
8
9     /**
10    * Creamos la respuesta.
11    * Por defecto es fallida.
12    */
13    $liked = FALSE;
14
15    /**
16    * Si el usuario esta logeado,
17    * los datos son validos y el id del usuario concuerda con el
    usuario actual,
18    * se crea o elimina el like.
19    */
20    if ($auth->isLogged()) {
21        if (isset($_POST['user_id']) && isset($_POST['share_id'])) {
22            if ($_POST['user_id'] == $arrays->value($auth-
23            >getCurrentUser(), 'user_id')) {
24                if ($database->isLikedBy($_POST['share_id'],
25                $_POST['user_id'])) {
```

```
24             $liked = !$database->unlike($_POST['share_id'],
$_POST['user_id']);
25         } else {
26             $liked = $database->like($_POST['share_id'],
$_POST['user_id']);
27         }
28     }
29 }
30 }
31
32 $database->close();
33
34 /**
35  * Enviamos la respuesta.
36  */
37 echo $liked;
38 ?>
```

login.php (public)

```
1 <?php
2     /**
3      * Incluimos el archivo de arranque de la aplicacion.
4      * Es muy importante por que contiene las funciones
5      * necesarias para que trabaje correctamente la aplicacion.
6      */
7     require_once '../boot.php';
8
9     /**
10    * Si el Usuario no esta autenticado.
11    * Verificamos sus credenciales,
12    * si son validas se redirecciona a la pagina principal.
13    */
14    if (!$auth->isLoggedIn()) {
15        if (isset($_POST['user_name']) && isset($_POST['password']))
16        {
17            $remember = isset($_POST['remember']);
18
19            if (!$auth->login($_POST['user_name'],
20                $_POST['password'], $remember)) {
21                $errors = TRUE;
22            } else {
23                $redirect->to('index.php');
24            }
25        }
26    }
```

```
25 ?>
26 <!doctype html>
27 <!-- Establecemos el lenguaje de la aplicacion -->
28 <html lang="<?php echo $SMM['lang']; ?>">
29 <head>
30     <!-- Establecemos la codificacion de caracteres -->
31     <meta charset="<?php echo $SMM['charset']; ?>">
32     <!-- No escalable -->
33     <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
34     <!-- Establecemos el titulo de la aplicacion -->
35     <title><?php echo $SMM['title']; ?></title>
36     <!-- Incluimos los estilos css -->
37     <?php foreach ($SMM['styles'] as $style): ?>
38         <link rel="stylesheet" href="<?php echo $style; ?>">
39     <?php endforeach; ?>
40 </head>
41 <body>
42     <!-- Contenedor principal -->
43     <div class="container">
44         <!-- Cabecera de la aplicacion -->
45         <div class="row">
46             <div class="col-xs-12">
47                 <?php include 'header.php'; ?>
48             </div>
49         </div>
```

```

50      <!-- Cuerpo de la aplicacion -->
51      <div class="row">
52          <!-- Menu de navegacion -->
53          <div class="col-xs-3">
54              <?php include 'sidebar.php'; ?>
55          </div>
56          <!-- Contenido -->
57          <div class="col-xs-9">
58              <!-- Shares -->
59              <form class="form-horizontal" action="login.php"
method="POST">
60                  <div class="panel panel-default">
61                      <div class="panel-heading">
62                          <h1 class="panel-title">Login</h1>
63                      </div>
64                      <div class="panel-body">
65                          <?php if (isset($errors)): ?>
66                              <span class="help-block alert alert-
danger">Usuario y/o contraseña invalidos.</span>
67                          <?php endif; ?>
68                          <!-- Usuario -->
69                          <div class="form-group">
70                              <label for="user_name" class="col-
sm-2 control-label">Usuario</label>
71                              <div class="col-sm-10">
72                                  <input type="text" class="form-
control" name="user_name">

```



```
73         </div>
74     </div>
75     <!-- Contraseña -->
76     <div class="form-group">
77         <label for="password" class="col-sm-
78 2 control-label">Contraseña</label>
79         <div class="col-sm-10">
80             <input type="password"
81 class="form-control" name="password">
82         </div>
83     </div>
84     <!-- Recordar usuario -->
85     <div class="form-group">
86         <div class="col-sm-8 col-sm-offset-
87 2">
88             <input type="checkbox"
89 name="remember"> Recordar
90         </div>
91     </div>
92     <div class="panel-footer">
93         <button type="submit" class="btn btn-
94 primary">Ingresar</button>
95     </div>
96 </form>
97 </div>
```

```
96         </div>
97     </div>
98
99     <!-- Incluimos el codigo javascript -->
100     <?php foreach ($SMM['scripts'] as $script): ?>
101         <script src="<?php echo $script; ?>"></script>
102     <?php endforeach; ?>
103
104     <?php
105         /**
106          * Incluimos el archivo para finalizar la ejecucion de la
107          aplicacion.
108          * Es muy importante para manejar la aplicacion en modo de
109          depuracion.
110          */
111         require_once '../stop.php';
112     ?>
113 </body>
114 </html>
```

logout.php (public)

```
1 <?php
2     /**
3      * Incluimos el archivo de arranque de la aplicacion.
4      * Es muy importante por que contiene las funciones
5      * necesarias para que trabaje correctamente la aplicacion.
6      */
7     require_once '../boot.php';
8
9     if ($auth->isLogged()) {
10         $auth->logout();
11     }
12
13     $redirect->to('welcome.php');
14 ?>
```

profile.php (public)

```
1 <?php
2     /**
3      * Incluimos el archivo de arranque de la aplicacion.
4      * Es muy importante por que contiene las funciones
5      * necesarias para que trabaje correctamente la aplicacion.
6      */
7     require_once '../boot.php';
8
9     /**
10    * Si esta autenticado.
11    */
12    if ($auth->isLogged()) {
13        /**
14        * Bandera para verificar si se actualizo el perfil.
15        */
16        $updated = FALSE;
17
18        /**
19        * Bandera para verificar si existen errores.
20        */
21        $errors = FALSE;
22
23        /**
24        * Validaciones para el formulario.
25        */
```

```

26     $validations = array(
27         'first_name' => '/^[a-zA-Zá-úÁ-Ú]+$/', // Solo
caracteres.
28         'last_name' => '/^[a-zA-Zá-úÁ-Ú]+$/', // Solo
caracteres.
29         'birthdate' => '/^([0-9]{4})-([0-9]{2})-([0-9]{2})$/',
// Solo Fechas.
30         'email' => '/^([a-zA-Z0-9])+([a-zA-Z0-9\._-])*@([a-zA-
Z0-9_-])+([a-zA-Z0-9\._-]+)$/', // Correo valido.
31     );
32
33     $currentUser = $auth->getCurrentUser();
34     $currentPassword = $currentUser['password'];
35
36     /**
37     * Validamos el nombre del usuario.
38     */
39     if (isset($_POST['first_name'])) {
40         $firstName = $_POST['first_name'];
41
42         if (preg_match($validations['first_name'], $firstName))
{
43             if (20 < strlen($firstName)) {
44                 $errors = TRUE;
45                 $_SESSION['form_errors']['first_name'] = 'Maximo
20 caracteres.';
46             } else {
47                 $updated = TRUE;

```

```
48         $currentUser['first_name'] = $firstName;
49     }
50     } else {
51         $errors = TRUE;
52         $_SESSION['form_errors']['first_name'] = 'Solo
caracteres [a-z], no puede estar vacia.';
53     }
54 }
55
56 /**
57  * Validamos el apellido del usuario.
58  */
59 if (isset($_POST['last_name'])) {
60     $lastName = $_POST['last_name'];
61
62     if (preg_match($validations['last_name'], $lastName)) {
63         if (20 < strlen($lastName)) {
64             $errors = TRUE;
65             $_SESSION['form_errors']['last_name'] = 'Maximo
20 caracteres.';
66         } else {
67             $updated = TRUE;
68             $currentUser['last_name'] = $lastName;
69         }
70     } else {
71         $errors = TRUE;
72         $_SESSION['form_errors']['last_name'] = 'Solo
```

```
caracteres [a-z], no puede estar vacia.';

73         }
74     }
75
76     /**
77     * Validamos la fecha de nacimiento.
78     */
79     if (isset($_POST['birthdate'])) {
80         $birthdate = $_POST['birthdate'];
81
82         if (preg_match($validations['birthdate'], $birthdate)) {
83             $updated = TRUE;
84             $currentUser['birthdate'] = $birthdate;
85         } else {
86             $errors = TRUE;
87             $_SESSION['form_errors']['birthdate'] = 'La fecha no
es valida.';
88         }
89     }
90
91     /**
92     * Validamos el correo.
93     */
94     if (isset($_POST['email'])) {
95         $email = $_POST['email'];
96     }
```

```
97         if (preg_match($validations['email'], $email)) {
98             $exist = $database->getUserWithEmail($email);
99             if ($exist && $exist['user_id'] !=
100 $currentUser['user_id']) {
101                 $errors = TRUE;
102                 $_SESSION['form_errors']['email'] = 'El correo
103 ya esta en uso.';
104             } else {
105                 $updated = TRUE;
106                 $currentUser['email'] = $email;
107             }
108         } else {
109             $errors = TRUE;
110             $_SESSION['form_errors']['email'] = 'El correo no es
111 valido.';
112         }
113     }
114
115     /**
116     * Validamos la nueva contraseña.
117     */
118     if (isset($_POST['new_password']) && 0 !=
119 strlen($_POST['new_password'])) {
120         if (isset($_POST['confirm_password'])) {
121             $newPassword = $_POST['new_password'];
122             $confirmPassword = $_POST['confirm_password'];
```



```
120         if (5 > strlen($newPassword) || 20 <
strlen($newPassword)) {
121             $errors = TRUE;
122             $_SESSION['form_errors']['new_password'] = 'La
contraseña tiene que tener entre 5 y 20 caracteres.';
123         } else {
124             if ($newPassword == $confirmPassword) {
125                 $updated = TRUE;
126                 $currentUser['password'] =
md5($newPassword);
127             } else {
128                 $errors = TRUE;
129                 $_SESSION['form_errors']['new_password'] =
'Las contraseñas no coinciden.';
130             }
131         }
132     } else {
133         $errors = TRUE;
134         $_SESSION['form_errors']['confirm_password'] =
'Confirma la nueva contraseña.';
135     }
136 }
137
138 /**
139  * Si se actualizo la informacion.
140  * Validamos la contraseña actual.
141  * Si es valida actualizamos la informacion del usuario.
142  */
```

```

143     if ($updated) {
144         if (isset($_POST['password'])) {
145             $password = $_POST['password'];
146
147             if ($currentPassword == md5($password)) {
148                 if (!$errors) {
149                     $database->updateUser($currentUser);
150
151                     $_SESSION['user'] = $currentUser;
152                 }
153             } else {
154                 $errors = TRUE;
155                 $_SESSION['form_errors']['password'] = 'La
contraseña no es valida..';
156             }
157         } else {
158             $errors = TRUE;
159             $_SESSION['form_errors']['password'] = 'Ingresa la
contraseña actual.';
160         }
161     }
162 }
163 ?>
164 <!doctype html>
165 <!-- Establecemos el lenguaje de la aplicacion -->
166 <html lang="<?php echo $SMM['lang']; ?>">
167 <head>

```

```
168     <!-- Establecemos la codificacion de caracteres -->
169     <meta charset="<?php echo $SMM['charset']; ?>">
170     <!-- No escalable -->
171     <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
172     <!-- Establecemos el titulo de la aplicacion -->
173     <title><?php echo $SMM['title']; ?></title>
174     <!-- Incluimos los estilos css -->
175     <?php foreach ($SMM['styles'] as $style): ?>
176         <link rel="stylesheet" href="<?php echo $style; ?>">
177     <?php endforeach; ?>
178 </head>
179 <body>
180     <!-- Contenedor principal -->
181     <div class="container">
182         <!-- Cabecera de la aplicacion -->
183         <div class="row">
184             <div class="col-xs-12">
185                 <?php include 'header.php'; ?>
186             </div>
187         </div>
188         <!-- Cuerpo de la aplicacion -->
189         <div class="row">
190             <!-- Menu de navegacion -->
191             <div class="col-xs-3">
192                 <?php include 'sidebar.php'; ?>
```

```

193         </div>
194         <!-- Contenido -->
195         <div class="col-xs-9">
196             <!-- Shares -->
197             <form class="form-horizontal" action="profile.php"
method="POST">
198                 <div class="panel panel-warning">
199                     <div class="panel-heading">
200                         <h1 class="panel-title">Editar Perfil:
<?php echo $arrays->value($auth->getCurrentUser(),
'user_name'); ?></h1>
201                     </div>
202                     <div class="panel-body">
203                         <div class="media">
204                             <div class="pull-left">
205                                 
206                             </div>
207                             <div class="media-body">
208                                 <!-- Nombre -->
209                                 <div class="form-group">
210                                     <label for="first_name"
class="col-sm-2 control-label">Nombre</label>
211                                     <div class="col-sm-10">
212                                         <input type="text"
class="form-control" name="first_name" value="<?php echo $arrays-
>value($auth->getCurrentUser(), 'first_name'); ?>">
213                                         <?php if

```

```

214         (isset($_SESSION['form_errors']['first_name'])): ?>
        <span class="help-
block alert alert-danger"><?php echo
$_SESSION['form_errors']['first_name']; ?></span>
215         <?php
unset($_SESSION['form_errors']['first_name']); ?>
216         <?php endif; ?>
217     </div>
218 </div>
219 <!-- Apellido -->
220 <div class="form-group">
221     <label for="last_name"
class="col-sm-2 control-label">Apellido</label>
222     <div class="col-sm-10">
223         <input type="text"
class="form-control" name="last_name" value="<?php echo $arrays-
>value($auth->getCurrentUser(), 'last_name'); ?>">
224         <?php if
(isset($_SESSION['form_errors']['last_name'])): ?>
225         <span class="help-
block alert alert-danger"><?php echo
$_SESSION['form_errors']['last_name']; ?></span>
226         <?php
unset($_SESSION['form_errors']['last_name']); ?>
227         <?php endif; ?>
228     </div>
229 </div>
230 <!-- Fecha de nacimiento -->
231 <div class="form-group">
232     <label for="birthdate"

```

```

class="col-sm-2 control-label">Fecha de nacimiento</label>
233         <div class="col-sm-10">
234             <input type="date"
class="form-control" name="birthdate" value="<?php echo $arrays-
>value($auth->getCurrentUser(), 'birthdate'); ?>">
235             <?php if
(isset($_SESSION['form_errors']['birthdate'])): ?>
236                 <span class="help-
block alert alert-danger"><?php echo
$_SESSION['form_errors']['birthdate']; ?></span>
237             <?php
unset($_SESSION['form_errors']['birthdate']); ?>
238             <?php endif; ?>
239         </div>
240     </div>
241     <!-- Correo -->
242     <div class="form-group">
243         <label for="email"
class="col-sm-2 control-label">Correo</label>
244         <div class="col-sm-10">
245             <input type="email"
class="form-control" name="email" value="<?php echo $arrays-
>value($auth->getCurrentUser(), 'email'); ?>">
246             <?php if
(isset($_SESSION['form_errors']['email'])): ?>
247                 <span class="help-
block alert alert-danger"><?php echo
$_SESSION['form_errors']['email']; ?></span>
248             <?php
unset($_SESSION['form_errors']['email']); ?>
249             <?php endif; ?>

```

```

250         </div>
251     </div>
252     <!-- Nueva contraseña -->
253     <div class="form-group">
254         <label for="new_password"
class="col-sm-2 control-label">Nueva contraseña</label>
255         <div class="col-sm-10">
256             <input type="password"
class="form-control" name="new_password">
257             <?php if
(isset($_SESSION['form_errors']['new_password'])): ?>
258                 <span class="help-
block alert alert-danger"><?php echo
$_SESSION['form_errors']['new_password']; ?></span>
259             <?php
unset($_SESSION['form_errors']['new_password']); ?>
260             <?php endif; ?>
261         </div>
262     </div>
263     <!-- Confirmar contraseña -->
264     <div class="form-group">
265         <label
for="confirm_password" class="col-sm-2 control-label">Confirmar
contraseña</label>
266         <div class="col-sm-10">
267             <input type="password"
class="form-control" name="confirm_password">
268             <?php if
(isset($_SESSION['form_errors']['confirm_password'])): ?>
269                 <span class="help-
block alert alert-danger"><?php echo

```

```

$_SESSION['form_errors']['confirm_password']; ?></span>
270                                     <?php
unset($_SESSION['form_errors']['confirm_password']); ?>
271                                     <?php endif; ?>
272                                 </div>
273                            </div>
274                            <!-- Contraseña actual -->
275                            <div class="form-group">
276                                <label for="password"
class="col-sm-2 control-label">Contraseña actual</label>
277                                <div class="col-sm-10">
278                                    <input type="password"
class="form-control" name="password">
279                                <?php if
(isset($_SESSION['form_errors']['password'])): ?>
280                                    <span class="help-
block alert alert-danger"><?php echo
$_SESSION['form_errors']['password']; ?></span>
281                                <?php
unset($_SESSION['form_errors']['password']); ?>
282                                    <?php endif; ?>
283                                </div>
284                            </div>
285                        </div>
286                    </div>
287                </div>
288                <div class="panel-footer">
289                    <button type="submit" class="btn btn-
warning">Guardar</button>

```



```
290         <a type="button" class="btn btn-default"
href="index.php">Cancelar</a>
291     </div>
292 </div>
293 </form>
294 </div>
295 </div>
296 </div>
297
298 <!-- Incluimos el codigo javascript -->
299 <?php foreach ($SMM['scripts'] as $script): ?>
300     <script src="<?php echo $script; ?>"></script>
301 <?php endforeach; ?>
302
303 <?php
304     /**
305      * Incluimos el archivo para finalizar la ejecucion de la
aplicacion.
306      * Es muy importante para manejar la aplicacion en modo de
depuracion.
307      */
308     require_once '../stop.php';
309 ?>
310 </body>
311 </html>
```

register.php (public)

```
1 <?php
2     /**
3      * Incluimos el archivo de arranque de la aplicacion.
4      * Es muy importante por que contiene las funciones
5      * necesarias para que trabaje correctamente la aplicacion.
6      */
7     require_once '../boot.php';
8
9     /**
10    * Si no esta autenticado.
11    */
12    if (!$auth->isLoggedIn() && isset($_POST['register'])) {
13        /**
14        * Bandera para verificar si existen errores.
15        */
16        $errors = FALSE;
17
18        /**
19        * Validaciones para el formulario.
20        */
21        $validations = array(
22            'user_name' => '/^[a-zA-Z]+$/', // Solo caracteres.
23            'first_name' => '/^[a-zA-Zá-úÁ-Ú]+$/', // Solo
caracteres.
24            'last_name' => '/^[a-zA-Zá-úÁ-Ú]+$/', // Solo
caracteres.
```

```

25         'birthdate' => '/^([0-9]{4})-([0-9]{2})-([0-9]{2})$/',
    // Solo Fechas.
26         'email' => '/^([a-zA-Z0-9])+([a-zA-Z0-9\._-])*@([a-zA-
Z0-9_-])+([a-zA-Z0-9\._-]+)$/', // Correo valido.
27     );
28
29     $values = array();
30
31     /**
32     * Validamos el nombre del usuario.
33     */
34     if (isset($_POST['first_name'])) {
35         $firstName = $_POST['first_name'];
36
37         if (preg_match($validations['first_name'], $firstName))
38     {
39             if (20 < strlen($firstName)) {
40                 $errors = TRUE;
41                 $_SESSION['form_errors']['first_name'] = 'Maximo
20 caracteres.';
42             } else {
43                 $values['first_name'] = $firstName;
44             }
45         } else {
46             $errors = TRUE;
47             $_SESSION['form_errors']['first_name'] = 'Solo
caracteres [a-z], no puede estar vacia.';
48         }

```

```
48         } else {
49             $errors = TRUE;
50             $_SESSION['form_errors']['first_name'] = 'Ingresa tu
nombre.';
51         }
52
53         /**
54          * Validamos el apellido del usuario.
55          */
56         if (isset($_POST['last_name'])) {
57             $lastName = $_POST['last_name'];
58
59             if (preg_match($validations['last_name'], $lastName)) {
60                 if (20 < strlen($lastName)) {
61                     $errors = TRUE;
62                     $_SESSION['form_errors']['last_name'] = 'Maximo
20 caracteres.';
63                 } else {
64                     $values['last_name'] = $lastName;
65                 }
66             } else {
67                 $errors = TRUE;
68                 $_SESSION['form_errors']['last_name'] = 'Solo
caracteres [a-z], no puede estar vacia.';
69             }
70         } else {
71             $errors = TRUE;
```

```
72         $_SESSION['form_errors']['last_name'] = 'Ingresa tu
apellido.';
73     }
74
75     /**
76     * Validamos la fecha de nacimiento.
77     */
78     if (isset($_POST['birthdate'])) {
79         $birthdate = $_POST['birthdate'];
80
81         if (preg_match($validations['birthdate'], $birthdate)) {
82             $values['birthdate'] = $birthdate;
83         } else {
84             $errors = TRUE;
85             $_SESSION['form_errors']['birthdate'] = 'La fecha no
es valida.';
86         }
87     } else {
88         $errors = TRUE;
89         $_SESSION['form_errors']['birthdate'] = 'Ingresa tu
fecha de nacimiento.';
90     }
91
92     /**
93     * Validamos el correo.
94     */
95     if (isset($_POST['email'])) {
```

```
96         $email = $_POST['email'];
97
98         if (preg_match($validations['email'], $email)) {
99             if ($database->getUserWithEmail($email)) {
100                 $errors = TRUE;
101                 $_SESSION['form_errors']['email'] = 'El correo
ya esta en uso.';
102             } else {
103                 $values['email'] = $email;
104             }
105         } else {
106             $errors = TRUE;
107             $_SESSION['form_errors']['email'] = 'El correo no es
valido.';
108         }
109     } else {
110         $errors = TRUE;
111         $_SESSION['form_errors']['email'] = 'Ingresa tu
correo.';
112     }
113
114     /**
115     * Validamos el nombre del usuario.
116     */
117     if (isset($_POST['user_name'])) {
118         $userName = $_POST['user_name'];
119     }
```

```
120         if (preg_match($validations['user_name'], $userName)) {
121             if (10 < strlen($userName)) {
122                 $errors = TRUE;
123                 $_SESSION['form_errors']['user_name'] = 'Maximo
10 caracteres.';
124             } else {
125                 if ($database->getUserWithUserName($userName)) {
126                     $errors = TRUE;
127                     $_SESSION['form_errors']['user_name'] = 'El
nombre de usuario ya esta en uso.';
128                 } else {
129                     $values['user_name'] = $userName;
130                 }
131             }
132         } else {
133             $errors = TRUE;
134             $_SESSION['form_errors']['user_name'] = 'Solo
caracteres [a-z], no puede estar vacia.';
135         }
136     } else {
137         $errors = TRUE;
138         $_SESSION['form_errors']['user_name'] = 'Ingresa tu
nombre de usuario.';
139     }
140
141     /**
142     * Validamos la contraseña.
```

```

143         */
144         if (isset($_POST['password']) && 0 !=
strlen($_POST['password'])) {
145             if (isset($_POST['confirm_password'])) {
146                 $password = $_POST['password'];
147                 $confirmPassword = $_POST['confirm_password'];
148
149                 if (5 > strlen($password) || 20 < strlen($password))
{
150                     $errors = TRUE;
151                     $_SESSION['form_errors']['password'] = 'La
contraseña tiene que tener entre 5 y 20 caracteres.';
152                 } else {
153                     if ($password == $confirmPassword) {
154                         $values['password'] = md5($password);
155                     } else {
156                         $errors = TRUE;
157                         $_SESSION['form_errors']['password'] = 'Las
contraseñas no coinciden.';
158                     }
159                 }
160             } else {
161                 $errors = TRUE;
162                 $_SESSION['form_errors']['confirm_password'] =
'Confirma la contraseña.';
163             }
164         } else {
165             $errors = TRUE;

```



```

166         $_SESSION['form_errors']['password'] = 'Ingresa tu
contraseña.';
167     }
168
169     /**
170      * Si no existen errores registramos al usuario y
redireccionamos a la pagina de login.
171     */
172     if (!$errors) {
173         $database->registerUser($values);
174         $redirect->to('login.php');
175     }
176 }
177 ?>
178 <!doctype html>
179 <!-- Establecemos el lenguaje de la aplicacion -->
180 <html lang="<?php echo $SMM['lang']; ?>">
181 <head>
182     <!-- Establecemos la codificacion de caracteres -->
183     <meta charset="<?php echo $SMM['charset']; ?>">
184     <!-- No escalable -->
185     <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
186     <!-- Establecemos el titulo de la aplicacion -->
187     <title><?php echo $SMM['title']; ?></title>
188     <!-- Incluimos los estilos css -->
189     <?php foreach ($SMM['styles'] as $style): ?>

```

```
190         <link rel="stylesheet" href="<?php echo $style; ?>">
191     <?php endforeach; ?>
192 </head>
193 <body>
194     <!-- Contenedor principal -->
195     <div class="container">
196         <!-- Cabecera de la aplicacion -->
197         <div class="row">
198             <div class="col-xs-12">
199                 <?php include 'header.php'; ?>
200             </div>
201         </div>
202         <!-- Cuerpo de la aplicacion -->
203         <div class="row">
204             <!-- Menu de navegacion -->
205             <div class="col-xs-3">
206                 <?php include 'sidebar.php'; ?>
207             </div>
208             <!-- Contenido -->
209             <div class="col-xs-9">
210                 <!-- Shares -->
211                 <form class="form-horizontal" action="register.php"
method="POST">
212                     <div class="panel panel-primary">
213                         <div class="panel-heading">
214                             <h1 class="panel-title">Registro</h1>
```

```

215         </div>
216         <div class="panel-body">
217             <!-- Indica que se envio el formulario
de registro -->
218             <input type="hidden" name="register">
219             <!-- Nombre -->
220             <div class="form-group">
221                 <label for="first_name" class="col-
sm-2 control-label">Nombre</label>
222                 <div class="col-sm-10">
223                     <input type="text" class="form-
control" name="first_name" value="<?php
echo !isset($_POST['first_name']) ? '' : $_POST['first_name']; ?>">
224                     <?php if
(isset($_SESSION['form_errors']['first_name'])): ?>
225                         <span class="help-block
alert alert-danger"><?php echo
$_SESSION['form_errors']['first_name']; ?></span>
226                     <?php
unset($_SESSION['form_errors']['first_name']); ?>
227                     <?php endif; ?>
228                 </div>
229             </div>
230             <!-- Apellido -->
231             <div class="form-group">
232                 <label for="last_name" class="col-
sm-2 control-label">Apellido</label>
233                 <div class="col-sm-10">
234                     <input type="text" class="form-
control" name="last_name" value="<?php

```

```

echo !isset($_POST['last_name']) ? '' : $_POST['last_name']; ?>">
235             <?php if
(isisset($_SESSION['form_errors']['last_name'])): ?>
236                 <span class="help-block
alert alert-danger"><?php echo
$_SESSION['form_errors']['last_name']; ?></span>
237             <?php
unset($_SESSION['form_errors']['last_name']); ?>
238             <?php endif; ?>
239         </div>
240     </div>
241     <!-- Fecha de nacimiento -->
242     <div class="form-group">
243         <label for="birthdate" class="col-
sm-2 control-label">Fecha de nacimiento</label>
244         <div class="col-sm-10">
245             <input type="date" class="form-
control" name="birthdate" value="<?php
echo !isset($_POST['birthdate']) ? '' : $_POST['birthdate']; ?>">
246             <?php if
(isisset($_SESSION['form_errors']['birthdate'])): ?>
247                 <span class="help-block
alert alert-danger"><?php echo
$_SESSION['form_errors']['birthdate']; ?></span>
248             <?php
unset($_SESSION['form_errors']['birthdate']); ?>
249             <?php endif; ?>
250         </div>
251     </div>
252     <!-- Correo -->

```

```

253         <div class="form-group">
254             <label for="email" class="col-sm-2
control-label">Correo</label>
255             <div class="col-sm-10">
256                 <input type="email" class="form-
control" name="email" value="<?php echo !isset($_POST['email']) ?
'' : $_POST['email']; ?>">
257                 <?php if
(isset($_SESSION['form_errors']['email'])): ?>
258                     <span class="help-block
alert alert-danger"><?php echo
$_SESSION['form_errors']['email']; ?></span>
259                 <?php
unset($_SESSION['form_errors']['email']); ?>
260                 <?php endif; ?>
261             </div>
262         </div>
263         <!-- Nombre de usuario -->
264         <div class="form-group">
265             <label for="user_name" class="col-
sm-2 control-label">Nombre de usuario</label>
266             <div class="col-sm-10">
267                 <input type="text" class="form-
control" name="user_name" value="<?php
echo !isset($_POST['user_name']) ? '' : $_POST['user_name']; ?>">
268                 <?php if
(isset($_SESSION['form_errors']['user_name'])): ?>
269                     <span class="help-block
alert alert-danger"><?php echo
$_SESSION['form_errors']['user_name']; ?></span>
270                 <?php

```

```
unset($_SESSION['form_errors']['user_name']); ?>
271         <?php endif; ?>
272     </div>
273 </div>
274 <!-- Contraseña -->
275 <div class="form-group">
276     <label for="password" class="col-sm-
277 2 control-label">Contraseña</label>
278     <div class="col-sm-10">
279         <input type="password"
280 class="form-control" name="password">
281         <?php if
282 (isset($_SESSION['form_errors']['password'])): ?>
283             <span class="help-block
284 alert alert-danger"><?php echo
285 $_SESSION['form_errors']['password']; ?></span>
286         <?php
287 unset($_SESSION['form_errors']['password']); ?>
288         <?php endif; ?>
289     </div>
290 </div>
<!-- Confirmar contraseña -->
<div class="form-group">
    <label for="confirm_password"
class="col-sm-2 control-label">Confirmar contraseña</label>
    <div class="col-sm-10">
        <input type="password"
class="form-control" name="confirm_password">
        <?php if
```

```

(isset($_SESSION['form_errors']['confirm_password'])): ?>
291         <span class="help-block
alert alert-danger"><?php echo
$_SESSION['form_errors']['confirm_password']; ?></span>
292         <?php
unset($_SESSION['form_errors']['confirm_password']); ?>
293         <?php endif; ?>
294     </div>
295 </div>
296 </div>
297 <div class="panel-footer">
298     <button type="submit" class="btn btn-
primary">Registrar</button>
299     <a type="button" class="btn btn-default"
href="index.php">Cancelar</a>
300 </div>
301 </div>
302 </form>
303 </div>
304 </div>
305 </div>
306
307 <!-- Incluimos el codigo javascript -->
308 <?php foreach ($SMM['scripts'] as $script): ?>
309     <script src="<?php echo $script; ?>"></script>
310 <?php endforeach; ?>
311
312 <?php

```

```
313         /**
314         * Incluimos el archivo para finalizar la ejecucion de la
315         * aplicacion.
316         * Es muy importante para manejar la aplicacion en modo de
317         * depuracion.
318         */
319         require_once '../stop.php';
320     ?>
321 </body>
322 </html>
```


share.php (public)

```
1 <?php
2     /**
3      * Incluimos el archivo de arranque de la aplicacion.
4      * Es muy importante por que contiene las funciones
5      * necesarias para que trabaje correctamente la aplicacion.
6      */
7     require_once '../boot.php';
8
9     if ($auth->isLogged()) {
10         if (isset($_POST['text'])) {
11             $text = $_POST['text'];
12
13             if (1 <= strlen($text)) {
14                 if (120 < strlen($text)) {
15                     $text = substr($text, 0, 120);
16                 }
17
18                 $database->share($arrays->value($auth-
19 >getCurrentUser(), 'user_id'), $text);
20             }
21         }
22
23         $redirect->to("board.php");
24     ?>
```

sidebar.php (public)

```
1 <ul class="nav nav-pills nav-stacked">
2   <li><a href="index.php">Inicio</a></li>
3   <?php if ($auth->isLoggedIn()): ?>
4       <li><a href="profile.php">Mi Perfil</a></li>
5       <li><a href="board.php">Mi Pizarra</a></li>
6   <?php endif; ?>
7 </ul>
```

welcome.php (public)

```
1 <?php
2     /**
3      * Incluimos el archivo de arranque de la aplicacion.
4      * Es muy importante por que contiene las funciones
5      * necesarias para que trabaje correctamente la aplicacion.
6      */
7     require_once '../boot.php';
8 ?>
9 <!doctype html>
10 <!-- Establecemos el lenguaje de la aplicacion -->
11 <html lang="<?php echo $SMM['lang']; ?>">
12 <head>
13     <!-- Establecemos la codificacion de caracteres -->
14     <meta charset="<?php echo $SMM['charset']; ?>">
15     <!-- No escalable -->
16     <meta name="viewport" content="width=device-width, initial-
17 scale=1.0, maximum-scale=1.0, user-scalable=no">
18     <!-- Establecemos el titulo de la aplicacion -->
19     <title><?php echo $SMM['title']; ?></title>
20     <!-- Incluimos los estilos css -->
21     <?php foreach ($SMM['styles'] as $style): ?>
22         <link rel="stylesheet" href="<?php echo $style; ?>">
23     <?php endforeach; ?>
24 </head>
25 <body>
```

```

25     <!-- Contenedor principal -->
26     <div class="container">
27         <!-- Cabecera de la aplicacion -->
28         <div class="row">
29             <div class="col-xs-12">
30                 <?php include 'header.php'; ?>
31             </div>
32         </div>
33         <!-- Cuerpo de la aplicacion -->
34         <div class="row">
35             <!-- Contenido -->
36             <div class="col-xs-12">
37                 <div class="jumbotron">
38                     <h1 class="text-center">¡Bienvenido a
ShareMyMind!</h1>
39                     <p class="text-center">
40                         <strong><?php echo $SMM['title']; ?></strong>
41                         es el lugar donde puedes compartir tus ideas
al mundo sin limitaciones.
42                     </p>
43                     <p>
44                         <a class="btn btn-primary btn-lg btn-block"
href="login.php">Ingresar</a>
45                         <a class="btn btn-default btn-lg btn-block"
href="register.php">Registrarse</a>
46                     </p>
47                     <p class="text-center"><a
href="index.php"><small>Ingresar a la pagina

```

```
principal</small></a></p>
48         </div>
49     </div>
50 </div>
51 </div>
52
53 <!-- Incluimos el codigo javascript -->
54 <?php foreach ($SMM['scripts'] as $script): ?>
55     <script src="<?php echo $script; ?>"></script>
56 <?php endforeach; ?>
57
58 <?php
59     /**
60      * Incluimos el archivo para finalizar la ejecucion de la
61      aplicacion.
62      * Es muy importante para manejar la aplicacion en modo de
63      depuracion.
64      */
65     require_once '../stop.php';
66 ?>
67 </body>
68 </html>
```

Glosario de funciones

FUNCION	DESCRIPCION	URL
array	Crea un array.	http://php.net/manual/es/function.array.php
array_push	Inserta uno o más elementos al final de un array.	http://php.net/manual/es/function.array-push.php
basename	Devuelve el componente de nombre de rastreo de la ruta.	http://php.net/manual/es/function.basename.php
ceil	Redondear fracciones hacia arriba.	http://php.net/manual/es/function.ceil.php
count	Cuenta todos los elementos de un array o en un objeto.	http://php.net/manual/es/function.count.php
date	Dar formato a la fecha/hora local.	http://php.net/manual/es/function.date.php
die	Equivalente a exit.	http://php.net/manual/es/function.die.php
dirname	Devuelve el directorio padre de la ruta.	http://php.net/manual/es/function.dirname.php
echo	Muestra una o más cadenas.	http://php.net/manual/es/function.echo.php
exit	Imprime un mensaje y termina el script actual.	http://php.net/manual/es/function.exit.php
header	Enviar encabezado sin formato HTTP.	http://php.net/manual/es/function.header.php
htmlentities	Convierte todos los caracteres aplicables a entidades HTML.	http://php.net/manual/es/function.htmlentities.php
implode	Une elementos de un array en un string.	http://php.net/manual/es/function.implode.php
include	La sentencia include incluye y evalúa el archivo especificado.	http://php.net/manual/es/function.include.php
is_numeric	Comprueba si una variable es un número o un string numérico.	http://php.net/manual/es/function.is-numeric.php
is_string	Comprueba si una variable es de tipo string.	http://php.net/manual/es/function.is-string.php
isset	Determina si una variable está definida y no es NULL.	http://php.net/manual/es/function.isset.php
md5	Calcula el hash md5 de un string.	http://php.net/manual/es/function.md5.php
mysql_close	Cierra una conexión de MySQL.	http://php.net/manual/es/function.mysql-close.php
mysql_connect	Abre una conexión al servidor MySQL.	http://php.net/manual/es/function.mysql-connect.php
mysql_escape_string	Escapa una cadena para ser usada en mysql_query.	http://php.net/manual/es/function.mysql-escape-string.php
mysql_fetch_assoc	Recupera una fila de	http://php.net/manual/es/function.mysql-fetch-

	resultados como un array asociativo.	assoc.php
mysql_free_result	Libera la memoria del resultado.	http://php.net/manual/es/function.mysql-free-result.php
mysql_query	Enviar una consulta MySQL.	http://php.net/manual/es/function.mysql-query.php
mysql_select_db	Seleccionar una base de datos MySQL.	http://php.net/manual/es/function.mysql-select-db.php
mysql_set_charset	Establece el conjunto de caracteres del cliente.	http://php.net/manual/es/function.mysql-set-charset.php
preg_match	Realiza una comparación con una expresión regular.	http://php.net/manual/es/function.preg-match.php
preg_replace	Realiza una búsqueda y sustitución de una expresión regular.	http://php.net/manual/es/function.preg-replace.php
realpath	Devuelve el nombre de la ruta absoluta canonizado.	http://php.net/manual/es/function.realpath.php
require_once	La sentencia require_once es idéntica a require excepto que PHP verificará si el archivo ya ha sido incluido y si es así, no se incluye (require) de nuevo.	http://php.net/manual/es/function.require-once.php
session_start	Crea una sesión o reanuda la actual basada en un identificador de sesión pasado mediante una petición GET o POST, o pasado mediante una cookie.	http://php.net/manual/es/function.session-start.php
session_destroy	Destruye toda la información registrada de una sesión.	http://php.net/manual/es/function.session-destroy.php
setcookie	Enviar una cookie.	http://php.net/manual/es/function.setcookie.php
str_shuffle	Reordena aleatoriamente una cadena.	http://php.net/manual/es/function.str-shuffle.php
strlen	Obtiene la longitud de un string.	http://php.net/manual/es/function strlen.php
strtolower	Convierte una cadena a minúsculas.	http://php.net/manual/es/function strtolower.php
substr	Devuelve parte de una cadena.	http://php.net/manual/es/function.substr.php
time	Devuelve la fecha Unix actual.	http://php.net/manual/es/function.time.php
trim	Elimina espacio en blanco (u otro tipo de caracteres) del inicio y el final de la cadena.	http://php.net/manual/es/function.trim.php
unset	Destruye una variable especificada.	http://php.net/manual/es/function.unset.php