

# Documentación Técnica resumida de la página web del proyecto TOMAS

## Resumen del Proyecto:

El proyecto consiste en una plataforma web para gestionar el acceso y la interacción de usuarios, especialmente orientada a la gestión de cargadores eléctricos y sus usuarios. El sistema permite a los usuarios registrarse, iniciar sesión, gestionar su perfil y acceder a una lista de cargadores cercanos mediante la geolocalización.

A continuación, se detallan los procesos realizados para completar el código del proyecto, incluyendo la estructura de la base de datos, la lógica de los archivos PHP y los aspectos de diseño de la interfaz web.

## 1. Estructura de la Base de Datos:

Para este proyecto, se asume que existe una base de datos MySQL que contiene las siguientes tablas principales:

### a. Tabla de Usuarios (usuarios)

- **id**: Identificador único del usuario (clave primaria).
- **nombre**: Nombre del usuario.
- **apellidos**: Apellidos del usuario.
- **fecha\_nacimiento**: Fecha de nacimiento del usuario.
- **telefono**: Número de teléfono del usuario.
- **email**: Correo electrónico del usuario (único).
- **contrasena**: Contraseña encriptada del usuario.

### b. Tabla de Cargadores (ubicaciones\_cargadores)

- **id**: Identificador único del cargador.
- **nombre**: Nombre del cargador.
- **direccion**: Dirección del cargador.
- **latitud**: Coordenada geográfica de latitud.
- **longitud**: Coordenada geográfica de longitud.
- **tipo\_cargador**: Tipo de cargador (por ejemplo, rápido, normal).

## 2. Procesos Realizados en los Archivos PHP

### a. conexion.php

Este archivo contiene la conexión a la base de datos. Se utiliza para conectar la aplicación PHP con la base de datos MySQL.

```
conexion.php > ...
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "root";
5  $dbname = "tomas";
6
7  // Crear conexión
8  $conn = new mysqli($servername, $username, $password, $dbname);
9
10 // Verificar conexión
11 if ($conn->connect_error) {
12     die("Conexión fallida: " . $conn->connect_error);
13 } else {
14     echo "Conexión exitosa a la base de datos.";
15 }
16 ?>
```

### b. acciones.php (Inicio de sesión)

Este archivo contiene la lógica para el inicio de sesión. Si el usuario introduce credenciales válidas, se inicia una sesión y se redirige a la página de su perfil.

- **Validación de usuario:** Se valida si el correo electrónico existe en la base de datos.
- **Verificación de contraseña:** Se compara la contraseña proporcionada con la contraseña almacenada en la base de datos (usando `password_verify()`).
- **Redirección:** Si las credenciales son correctas, se redirige al usuario a su área reservada.

### c. accionregistro.php (Registro de usuario)

Este archivo gestiona el registro de un nuevo usuario. Se verifica si el correo electrónico ya está registrado antes de insertarlo en la base de datos.

- **Verificación de correo electrónico:** Si el correo ya existe, se muestra un mensaje de error.

- **Inserción en la base de datos:** Si el correo no está registrado, se insertan los datos del usuario en la base de datos, incluyendo la contraseña encriptada.

```
2 // Comprobar si se envió el formulario mediante POST
3 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
```

#### d. micuenta.php (Página de inicio de sesión)

Este archivo contiene el formulario de inicio de sesión. Si el usuario está logueado, se le da la bienvenida y se le redirige a su área reservada. Si no, se muestra un formulario para ingresar las credenciales.

#### e. header.php (Barra de navegación)

El archivo header.php contiene la barra de navegación, que incluye enlaces a la página principal, el inicio de sesión y el perfil del usuario. También muestra un mensaje de bienvenida si el usuario está logueado y ofrece la opción de cerrar sesión.

#### f. areareservada.php (Área reservada)

Este archivo muestra el perfil del usuario una vez que ha iniciado sesión. Presenta los datos personales del usuario y proporciona enlaces para editar su perfil o cerrar sesión.

- **Comprobación de sesión:** Si el usuario no está logueado, se le redirige a la página de inicio de sesión.
- **Datos del usuario:** Se extraen los datos del usuario desde la sesión activa.

```
4 // Verificar si el usuario está logueado
5 if (!isset($_SESSION['usuario_id'])) {
6     // Si no está logueado, redirigir al login
7     header("Location: micuenta.php");
8     exit();
9 }
```

#### g. logout.php (Cerrar sesión)

Este archivo destruye la sesión activa y redirige al usuario a la página de inicio de sesión.

```

🐛 logout.php
1  <?php
2  session_start();
3  session_unset(); // Elimina todas las variables de sesión
4  session_destroy(); // Destruye la sesión
5
6  header("Location: micuenta.php"); // Redirige al login
7  exit();
8  ?>

```

#### h. getcharger.php (Carga de cargadores cercanos)

Este archivo obtiene los cargadores cercanos a la ubicación del usuario utilizando las coordenadas (latitud y longitud). La consulta SQL calcula la distancia entre el usuario y cada cargador y devuelve aquellos que se encuentran dentro de un radio de 50 km.

```

21
22 // Consulta SQL para obtener los cargadores cercanos dentro de un radio de 50 km
23 $sql = "
24     SELECT id, nombre, direccion, latitud, longitud, tipo_cargador,
25     (6371 * acos(cos(radians($lat)) * cos(radians(latitud)) * cos(radians(longitud) - radians($lng)) +
26     sin(radians($lat)) * sin(radians(latitud)))) AS distance
27     FROM ubicaciones_cargadores
28     HAVING distance < 50
29     ORDER BY distance
30 ";

```

### 3. Interfaz de Usuario (Frontend)

#### a. Diseño con Bootstrap 5

La interfaz se construyó utilizando **Bootstrap 5** para asegurar un diseño responsivo y atractivo. El uso de componentes como las tarjetas (card), los formularios (form), los botones (btn), y la barra de navegación (navbar) mejora la experiencia del usuario.

#### b. Página de Inicio de Sesión (micuenta.php)

El formulario de inicio de sesión se construyó utilizando las clases de Bootstrap para campos de entrada, botones y alertas de error. Si el usuario no está logueado, se le presenta un formulario de inicio de sesión, de lo contrario, se muestra un mensaje de bienvenida.

### c. Página de Registro (registro.php)

El formulario de registro permite a los usuarios ingresar su nombre, apellidos, fecha de nacimiento, teléfono, correo electrónico y contraseña. Se valida que el correo electrónico no esté registrado previamente en la base de datos antes de permitir el registro.

### d. Página de Perfil de Usuario (areareservada.php)

Esta página muestra el perfil del usuario con la posibilidad de editar los datos personales o cerrar sesión. El diseño es sencillo y centrado en el usuario.

## 4. Mejoras y Seguridad

- **Validación de Entradas:** Se implementó una validación básica en el lado del servidor para los formularios de registro e inicio de sesión.
- **Contraseña Segura:** Las contraseñas se almacenan de manera segura utilizando el algoritmo de encriptación **BCRYPT** (`password_hash()` y `password_verify()`).
- **Prevención de Inyección SQL:** Se utilizaron consultas preparadas para evitar vulnerabilidades de inyección SQL.
- **Protección XSS:** Se utilizaron funciones como `htmlspecialchars()` para evitar la ejecución de código malicioso en las páginas web.

## 5. Conclusión

Este proyecto es una plataforma web completa que permite la gestión de usuarios y cargadores eléctricos. Los usuarios pueden registrarse, iniciar sesión, ver su perfil y acceder a cargadores cercanos. Se utilizó una arquitectura segura con PHP, MySQL y Bootstrap 5, y se implementaron medidas de seguridad como validaciones y encriptación de contraseñas.

Este sistema es escalable y puede mejorarse agregando más funcionalidades, como la edición de perfiles, la visualización de historial de carga, y la integración con servicios de geolocalización más avanzados.