

DWES Proyecto PHP

Plataforma web montañera con PHP y MySQL



Francisco José García Egea

ÍNDICE

Tarea 1.1: Estructura inicial del proyecto.....	2
Tarea 1.2: Sistema de registro de usuarios.....	2
Tarea 1.3: Sistema de login y sesiones.....	4
Tarea 1.4: Página principal y navegación.....	6
Tarea 1.5: Creación de rutas con galería de fotos asociada.....	8

Leer README.md

Tareas del proyecto:

Tarea 1.1: Estructura inicial del proyecto

Objetivo: Crear la estructura de carpetas y archivos base.

Subir a github:

- Estructura de carpetas completas
- Archivos básicos creados
- Header y footer con HTML básico
- Archivo de configuración inicial (README.md)

Enlace al repositorio de GitHub:

https://github.com/Ffran-g/DWES_PROYECTO_PHP.git

Tarea 1.2: Sistema de registro de usuarios

Objetivo: Crear formulario de registro con validación.

Funcionalidades:

- Formulario con campos: username, email, password, confirm_password, nivel experiencia, especialidad, provincia
- Validación de campos obligatorios
- Validación formato email
- Validación coincidencia contraseñas
- Validación longitud mínima contraseña
- Mensajes de error específicos
- Almacenamiento temporal en array (sin BD aún)

Archivo: register.php

Registrarse con nosotros

Nombre de usuario
Fran Garcia

Email
nombre@ejemplo.com

Nunca compartiremos tu correo con nadie.

Contraseña

Confirmar Contraseña

Nivel
Selecciona tu nivel

Disciplinas

- Ferratas
- Escalada
- Rutas

Provincia
Selecciona tu provincia

Registrarme

Home Features Pricing FAQs About

© 2025 Company, Inc

Limpiar array de usuarios

El usuario se guarda en un array de sesión (`$_SESSION['user']`) en el mismo archivo `register.php`.

(También hay creada una variable `$message`, con mensajes personalizados dependiendo de si no rellenas algún campo, si el email no es correcto o si la contraseña no es valida)

Dentro de este array se guardan los datos introducidos por el usuario en el formulario, con el método POST.

Todos los datos son obligatorios para la correcta creación del usuario, lo cual se verifica comprobando si los campos están vacíos.

Antes de que los datos sean guardados en el array de sesión, son validados (también en el mismo archivo, menos el email y la contraseña que tienen su propia función de verificación en `function.php`).

Se comprueba si el email es correcto (nombre@algo.com), si la contraseña cumple con los requisitos mínimos y si la contraseña introducida y su confirmación coinciden.

Una vez visto todo eso, los datos se guardan en el array de sesión.

Tarea 1.3: Sistema de login y sesiones

Objetivo: Implementar autenticación básica con sesiones.

Funcionalidades:

- Formulario de login (username/email, password)
- Validación de credenciales contra array temporal
- Creación de sesión de usuario
- Página de perfil protegida
- Botón de logout
- Redirecciones apropiadas

Archivos: login.php, logout.php, profile.php, includes/auth_check.php

The screenshot shows a login form titled "Iniciar sesión" (Start session). It has fields for "Usuario o Email" (User or Email) and "Contraseña" (Password), both with placeholder text. There is a "Recuérdame" (Remember me) checkbox. A large orange "Entrar" (Enter) button is at the bottom. Below it, a link says "¿No tienes una cuenta con nosotros? REGISTRATE". At the top, there's a logo for "Mountain Climbing" and links for "Login" and "Sign-up". At the bottom, there are navigation links for "Home", "Features", "Pricing", "FAQs", and "About", along with a copyright notice and a red "Limpiar array de usuarios" (Clear user array) button.

Las verificaciones del login se hacen en auth_check.php.

En orden (por como está escrito el código), lo primero creado son unas cookies que funcionan con el botón de “Recuérdame”, si no se pulse, no se guardarán en ese campo los datos introducidos (el nombre de usuario o el email).

También hay un array de errores (creado en la sesión \$login_error = \$_SESSION['login_error'] ?? '') que si no existe, se crea.

En el auth_check.php, se comprueba si existe el usuario comparando el nombre de usuario o email introducido y la contraseña con el array de sesión de usuarios. Si

coincide, se marcan un booleano como true (en caso contrario, lanzara un login_error indicando que el usuario o contraseña son incorrectos).

The screenshot shows the 'Perfil de Usuario' (User Profile) page for a user named 'Fran'. At the top, there's a navigation bar with a mountain icon, the text 'Mountain Climbing', and links for 'Rutas', 'Ferratas', 'Escalada', and 'Galería'. On the right, there's a user icon, the name 'Fran', and a 'Cerrar sesión' (Logout) button. Below the navigation is a large circular placeholder image for the user's profile picture, with the text 'Información Personal' (Personal Information) below it. The profile information is listed in a table:

Nombre:	Fran
Correo electrónico:	fran@ejemplo.com
Nivel:	Amateur
Disciplinas:	Ferratas
Provincia:	Huesca

Below the table, it says 'Rutas creadas: 0' with a 'Ver lista de rutas' (View route list) button. At the bottom of the profile box are two buttons: 'Editar Perfil' (Edit Profile) and 'Cerrar Sesión' (Logout). At the very bottom of the page, there's a footer with links for 'Home', 'Features', 'Pricing', 'FAQs', 'About', and a copyright notice: '© 2025 Company, Inc'.

Pantallazo de profile.php

La imagen que se ve de usuario es la que he puesto por defecto.

Lo primero, es que el perfil está bloqueado, si no existe una sesión iniciada, no puedes acceder a el. (Esto lo compruebo con un !isset(\$_SESSION['user']), si no hay usuario, le redirigirá al login.php para que intente iniciar sesión).

Tanto para la foto, como para el resto de datos que se ven en la imagen (nombre, email, nivel ...) los muestro con 'htmlspecialchars', mostrando así de forma segura los datos del usuario (ya que convierte los caracteres especiales en sus formas codificadas) previendo que si se ha ingresado código en alguno de los campos, que el navegador lo ejecute.

Aunque todos los campos son obligatorios, profile.php tiene una comprobación por cada campo (a excepción del de usuario y email). <?php if (isset(\$user['campo'])) ?> que comprueba si hay algún valor y lo muestra. Como son campos obligatorios siempre estarán, pero no está demás recorrelos.

Tarea 1.4: Página principal y navegación

Objetivo: Crear la página de inicio con navegación dinámica.

Funcionalidades:

- Menú diferente para usuarios logueados/no logueados
- Mostrar nombre de usuario en header si está logueado
- Página de bienvenida con información del sitio
- Enlaces a diferentes secciones

Archivos: index.php, actualizar includes/header.php

The screenshot shows the homepage of a website for mountain climbing. At the top, there is a dark purple header bar with a logo of a mountain peak and the text "Mountain Climbing". To the right of the logo are two buttons: "Login" (in a blue box) and "Sign-up" (in a yellow box). Below the header, the main content area has a light gray background. The title "Explora el Mundo Vertical" is centered at the top of the content area in a large, bold, orange font. Below the title, there is a brief welcome message: "Bienvenido a nuestra comunidad de aventureros. Aquí encontrarás toda la información, rutas y consejos sobre ferratas, escalada y senderismo. ¡Prepárate para descubrir nuevas alturas y horizontes!" In the center of the page, there are three rounded rectangular boxes arranged horizontally. The first box is titled "Ferratas" and contains text about via ferratas. The second box is titled "Escalada" and contains text about climbing. The third box is titled "Rutas" and contains text about routes. At the bottom of the page, there is a footer bar with links: "Home", "Features", "Pricing", "FAQs", and "About". Below the footer, there is a copyright notice: "© 2025 Company, Inc" and a red button labeled "Limpiar array de usuarios".

Pantallazo de index.php sin sesión iniciada.



Explora el Mundo Vertical

Bienvenido a nuestra comunidad de aventureros. Aquí encontrarás toda la información, rutas y consejos sobre ferratas, escalada y senderismo.
¡Prepárate para descubrir nuevas alturas y horizontes!

Ferratas

Las vías ferratas son rutas equipadas con cables, grapas y escalerillas de acero que permiten ascender paredes rocosas con seguridad. Perfectas para quienes buscan una mezcla de adrenalina y paisaje.

[Lista de ferratas](#)

[Crear ferrata](#)

Escalada

La escalada deportiva es una disciplina que combina técnica, fuerza y concentración. Desde muros artificiales hasta rutas naturales, la escalada es una forma única de superarla física y mentalmente.

[Lista de escalada](#)

[Crear escalada](#)

Rutas

Las rutas de senderismo te conectan con la naturaleza a través de paisajes únicos. Descubre itinerarios adaptados a todos los niveles, desde paseos tranquilos hasta travesías exigentes.

[Lista de rutas](#)

[Crear ruta](#)

[Home](#) [Features](#) [Pricing](#) [FAQs](#) [About](#)

© 2025 Company, Inc

[Limpiar array de usuarios](#)

Pantallazo de index.php con sesión iniciada.

Como se aprecia en los pantallazos. Al tener una sesión iniciada el header cambia:

- Añade el nombre del usuario (que incluye un link para ir a su perfil)
- Cambia los botones de Sing-up y Login por el de Cerrar Sesión
- También se muestra una lista con los diferentes enlaces a las páginas.
- En el index se añaden botones para Ver o Crear nuevas rutas, ferratas o escaladas.

Esto se hace comprobando, en functions.php, si hay algún usuario conectado.

El motivo por el que en las capturas del punto 1.3 también se ve el header cambiado en profile.php es porque las estuve haciendo casi a la par y modifique primero el header.php antes de ponerme con profile.php.

Tarea 1.5: Creación de rutas con galería de fotos asociada

Objetivo: Diseñar e implementar un formulario completo para añadir rutas de senderismo, permitiendo opcionalmente adjuntar fotografías relacionadas con cada ruta.

Funcionalidades:

Formulario con campos principales de la ruta

- Nombre de la ruta
- Dificultad (select: fácil, moderada, difícil, muy difícil)
- Distancia en km
- Desnivel positivo
- Duración estimada (horas)
- Provincia (select con provincias españolas)
- Época recomendada (checkboxes: primavera, verano, otoño, invierno)
- Descripción (textarea)
- Nivel técnico (1-5)
- Nivel físico (1-5)

Validaciones y almacenamiento

- Validación completa de todos los campos
- Posibilidad de subir una o varias imágenes asociadas a la ruta
- Validación del tipo de archivo (jpg, jpeg, png)
- Validación del tamaño máximo (2MB)
- Renombrado seguro de archivos
- Almacenamiento en carpeta /uploads/photos/
- Almacenamiento temporal de la información en arrays o sesiones
- Listado simple de rutas creadas con miniaturas de sus fotos asociadas

Archivos

- routes/create.php
- routes/list.php
- uploads/photos/

The screenshot shows a 'Crear nueva ruta' (Create new route) form. The fields include:

- Nombre de la ruta (Route name)
- Dificultad (Difficulty) with a dropdown menu: 'Selecciona dificultad'
- Distancia (km) (Distance km)
- Desnivel positivo (m) (Positive elevation m)
- Duración estimada (horas) (Estimated duration hours)
- Provincia (Province) with a dropdown menu: 'Selecciona provincia'
- Época recomendada (Recommended season):
 - Primavera (Spring)
 - Verano (Summer)
 - Otoño (Autumn)
 - Invierno (Winter)
- Descripción (Description)
- Nivel técnico (1-5) (Technical level 1-5)
- Nivel físico (1-5) (Physical level 1-5)
- Fotos de la ruta (Photos of the route):
 - Elegir archivos (Select files)
 - Ningún archivo seleccionado (No file selected)
 - JPG, JPEG o PNG. Máx 2MB por archivo. (File types: JPG, JPEG or PNG. Max 2MB per file.)
- A large orange 'Crear ruta' (Create route) button.

Este es el formulario con todos los campos requeridos en la práctica para la creación de las rutas.

Funciona prácticamente igual al formulario de registro, con ligeros cambios:

Como este formulario debe recoger imágenes (no obligatorias). Lo primero que hace (después de comprobar que los demás campos no están vacíos) es comprobar el array de imágenes para ver si hay alguna subida, en caso de que no la haya, le asignara una por defecto como miniatura.

Pero si hay alguna imagen: Se recorren con un foreach, con un valor \$key (numero en el array) y un \$name (nombre del archivo de imagen).

Se guardan los siguientes datos de la/las imagen/es con \$_FILES:

- \$tmpName = \$_FILES['fotos']['tmp_name'][\$key]; Una ruta temporal hasta que se le reasigne.
- \$size = \$_FILES['fotos']['size'][\$key]; El tamaño en bytes.

- \$type = mime_content_type(\$tmpName); Usando la ruta temporal, se lee el tipo de archivo que es (si es image, jpeg).
- \$ext = strtolower(pathinfo(\$name, PATHINFO_EXTENSION)); Esto nos guarda la extensión.

Después se valida el tipo de archivo que es ('jpg', 'jpeg' o 'png')

Seguido de una confirmación de tamaño \$size > 2 * 1024 * 1024.

Si cualquiera de los casos anteriores no es valido, se muestra el error correspondiente y se para.

Solo una vez todo es valido, se le asigna un nombre único y se guarda en la carpeta /uploads/photos.