



# Mountain Connect

Lili Wu  
CPIFP Los Enlaces  
2 DAWV  
Entorno Servidor  
16/11/2025

# Índice

<b>Introducción.....</b>	<b>2</b>
1. Requisitos.....	3
2. Estructura Mountain-Connect.....	4
3. Funcionamiento de la página.....	5
4. Pruebas.....	11
5. Conclusión.....	13

# Introducción

Con el proyecto Mountain-Connect tenemos el objetivo de que a todas las personas que le gustan el senderismo, la escalada, conectar con la naturaleza, pueda acceder y tener sus rutas y compartir rutas que ya han realizado para que otros usuarios puedan ir y disfrutar de la naturaleza. Esta web está dirigida para todas las personas amantes de la naturaleza y conocer su entorno haciendo escalada, senderismo, etc. Gracias a esta web, todas las personas que le gustan este ámbito, pueden conocer nuevos lugares, gracias a las personas que comparten sus rutas y sus experiencias. Las principales funcionalidades que tiene es conectar con otros usuarios del mismo hobby.

[GITHUB REPOSITORIO MOUNTAIN-CONNECT](#)

## 1. Requisitos.

### 1.1 Requisitos funcionales

RF1: El usuario puede registrarse. Añadiendo (nombre, correo, experiencia, especialidad, provincia y contraseña )

RF2: El usuario puede iniciar sesión

RF3: El usuario logueado puede crear rutas

RF4: El usuario logueado puede subir imágenes

RF5: El usuario logueado puede ver su perfil

RF6: El usuario logueado puede cerrar sesión

RF7: El usuario logueado puede ver la lista de rutas que hay creadas.

### 1.2 Requisitos no funcionales

RNF1: Usabilidad, interfaz intuitiva

RNF2: Validación de datos del usuario

RNF3: Estructura del proyecto en las carpetas

RNF4: Uso de sesiones en PHP

RNF5: Rendimiento (que vaya rápido)

## 2. Estructura Mountain-Connect

/assets → En esta carpeta contiene subcarpetas que son: css, images, js

- css para el estilo de la página de php.
- images para las imágenes de la página web de php.
- js para que sea una página interactiva (no se ha usado en el proyecto.)

/config

- no se ha usado.

/includes

- auth\_check.php → se ha usado para ver si está o no la sesión activa.
- footer.php → El pie de la página en la que solo salen los derechos reservados.
- functions.php → no se ha usado ya que se han usado las funciones dentro del código php.
- header.php → para tener el encabezado de la página php. También comprueba si el usuario está logueado, para mostrar el perfil del usuario o por otro lado, que solo aparezcan los botones de iniciar sesión y registrarse

/public

- climbing es una carpeta que dentro tiene para crear una vía de escalada, la vía de escalada lo puede crear todo el mundo, seas usuario o no seas. También aparece el archivo de list.php que sirve para listar la vía de escalada que han creado los usuarios y no usuarios.
- ferratas → no se ha usado porque sería lo mismo que climbing.
- photos → no se ha usado.
- routes→ create.php sirve para crear una ruta. Y list, para listar la ruta creada. Los otros archivos no se han usado.
- index.php es la página inicial.
- login.php es la página para loguearse, una vez te hayas registrado o tengas cuenta ya creada.
- logout.php es el archivo que inicia la sesión, elimina las variables de la sesión y destruye la sesión una vez cierres sesión.
- profile.php → es la página del perfil, una vez que te logueas, te dirigirá a esa página.
- register.php→ es la página donde haces el registro.

/uploads

- photos → es donde se guardarán todas las fotos que se añaden en el momento de crear la ruta.
- profile → no usado.

### 3. Funcionamiento de la página

He creado el código de register.php en la que tiene un formulario dentro del html, el formulario recogerá los datos del usuario, y lo enviará de manera segura (al usar el POST), cada campo span MUESTRA UN ERROR.

En el nivel de experiencia, en caso de que haya un error en algún campo anterior, siga seleccionado el nivel de experiencia, y no se borre. Con el isset comprueba si existe y con el "selected". Se mantiene el valor seleccionado. Al igual pasa con la provincia seleccionada.

```
<option value="Principiante" <?= (isset($experiencia) && $experiencia === "Principiante") ? "selected" : "" ?>>Principiante</option>
```

El código register.php tiene todas las validaciones. Usando el método de la caja blanca.

Validaciones:

1. Evitar usuarios repetidos, es válido crear un usuario con el nombre de usuario Lili y lili, sin embargo, una vez creado, si vuelves a intentar crear un usuario Lili (con los mismos caracteres (mayusculas y minusculas)) no te dejará crearlo y te mostrará un mensaje de error.

```
foreach ($_SESSION['users'] as $user){
    if ($user['username'] === $username) {
        $errors['username'] = "El nombre del usuario ya está en uso.";
        break;
    }
}
```

2. Evitar correos repetidos, en este punto, da igual mayusculas o minusculas, si pones (a@gmail.com y A@GMAIL.COM) te lo pillara todo en minúsculas, por lo que si hay un usuario con ese correo ya creado, no te permitirá crearlo, por lo que dará error

```
if (strtolower ($user['email']) === strtolower($email)) {
    $errors['email']= "El email ya está registrado / en uso";
    break;
}
```

3. Obligatorio todos los campos, no se puede quedar vacío ningún campo. Por ejemplo: Si el campo de email está vacío, manda un error.

```
if (empty($email)) {
    $errors['email'] = "El email es obligatorio.";
```

4. Formato de correo electrónico, php tiene su filtro de validación de correo electrónico.

```
elseif (!filter_var($email, FILTER_VALIDATE_EMAIL))
```

5. La contraseña tiene que tener más de 8 dígitos, strlen sirve para leer la longitud de una cadena de texto

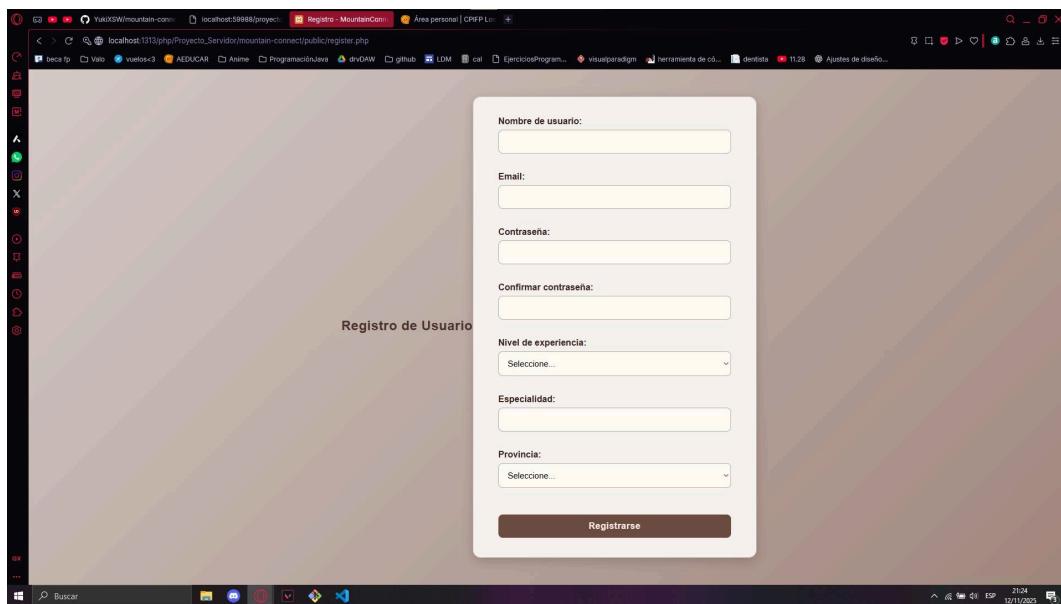
```
} elseif (strlen($password) < 8) {
    $errors['password'] = "La contraseña debe tener mínimo 8
caracteres."; # Si la contraseña tiene menos de 8 caracteres mostramos el
error
}
```

6. La contraseña confirmada tiene que coincidir con la contraseña

```
$password !== $confirm_password
```

Si no coincide envia un error

Aquí te presento la página de register.php



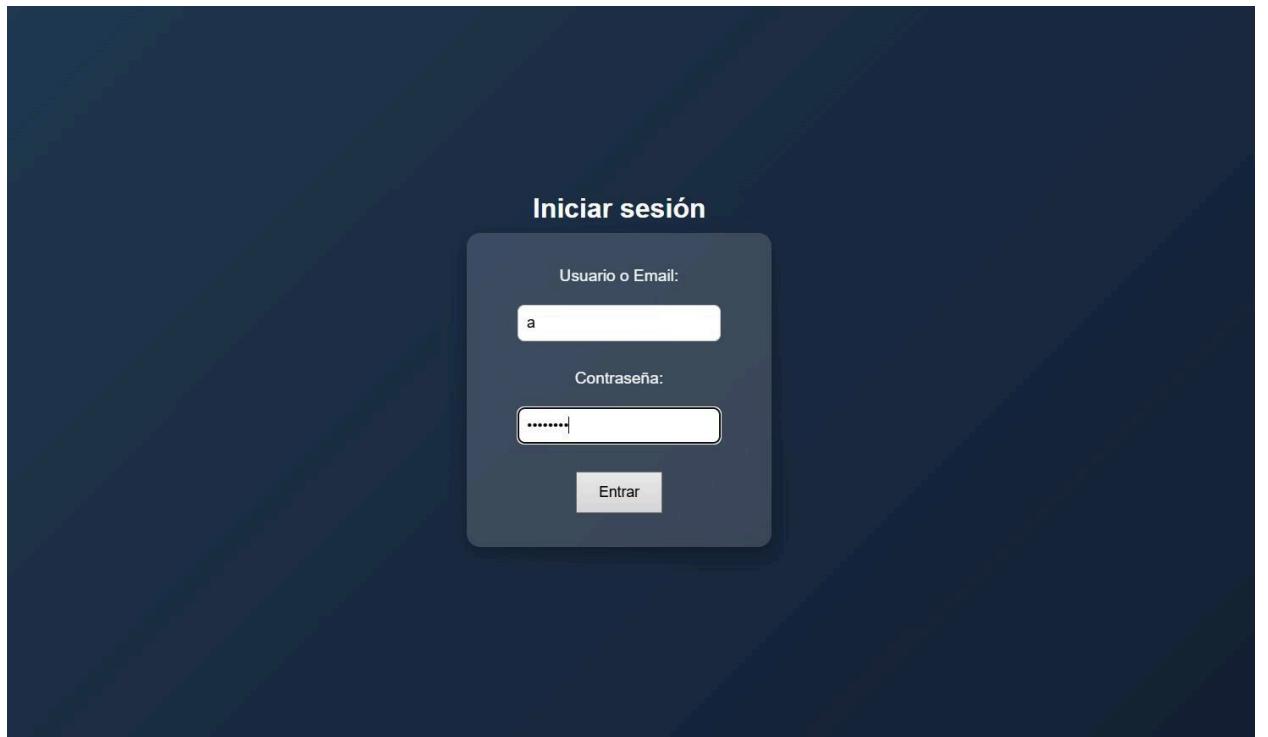
Al darle a registrar te tiene que llevar a la página de login.php.

En la que me he encontrado problemas como por ejemplo. Me creaba bien los usuarios, sin embargo a la hora de hacer login, me daba error en las credenciales, y en la única que no me daba error era en la primera cuenta creada. El problema se situaba en el array temporal. Guardé el array con valores y por ello me daba error.

```
$users_registered = $_SESSION['users'] ?? [];
```

Aquí guardas los usuarios que ya se han registrado.

En la página de login.php, tiene que buscar si el usuario existe, y si no existe, mandar un error. Al igual que manda un error si la contraseña está incorrecta. Sin embargo, si existe, directamente te manda a la página de profile.php.



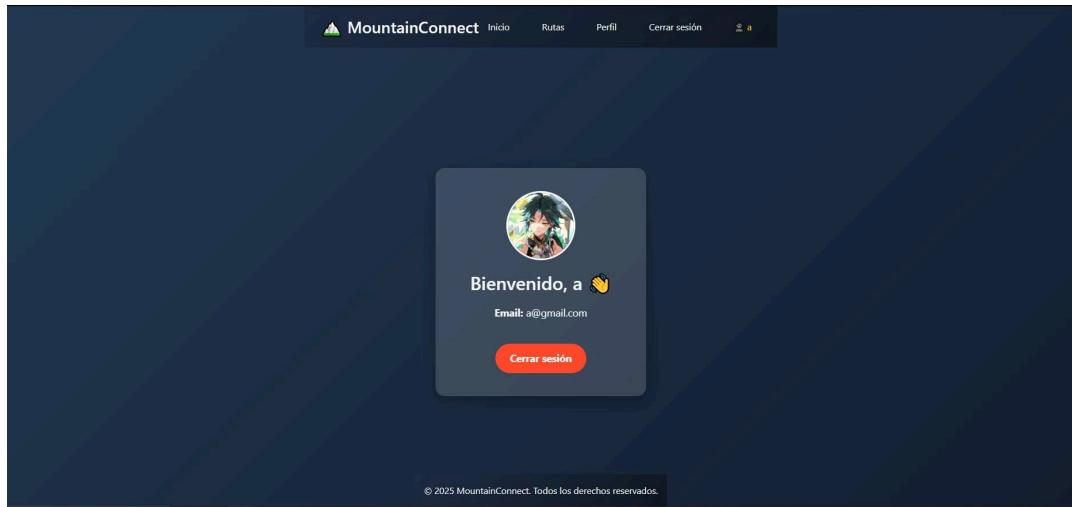
En la página de profile.php hemos conectado header de includes *includes/header.php*

```
<?php include '../includes/header.php'; ?>
```

Y el footer \_includes/footer.php\_

```
<?php include '../includes/footer.php'; ?>
```

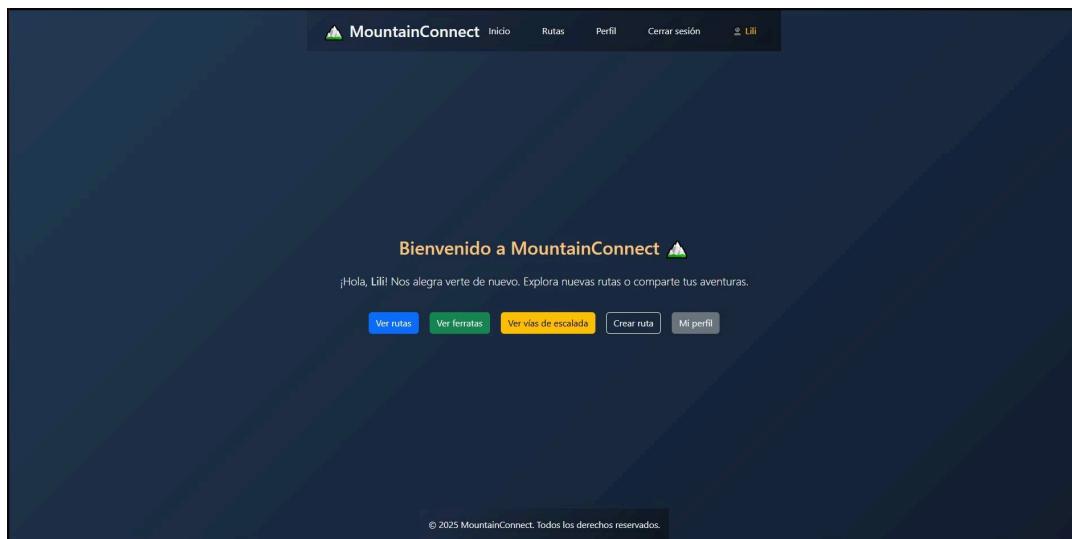
Y código html



Estando en la página de profile.php, en el header, tienes 4 opciones, la primera te lleva a la página principal que es /public/index.php.

La segunda opción te lleva al listado de rutas que es /public/routes/list.php, en caso de que haya rutas creadas te aparecerá la imagen o solo la información de la ruta. En caso de que no haya, también tienes la opción de crear una ruta. Accediendo a Inicio y en el cuerpo te aparecerán las opciones de **Ver rutas**, que es la misma opción que Rutas en el header, pero en el penúltimo botón aparece **Crear ruta** y ahí podréis crear la ruta que hayas hecho y quieras compartirlo para otras personas. Al lado de la opción de Rutas del encabezado, está la opción de ver tu perfil o cerrar sesión para cerrar sesión de la cuenta.

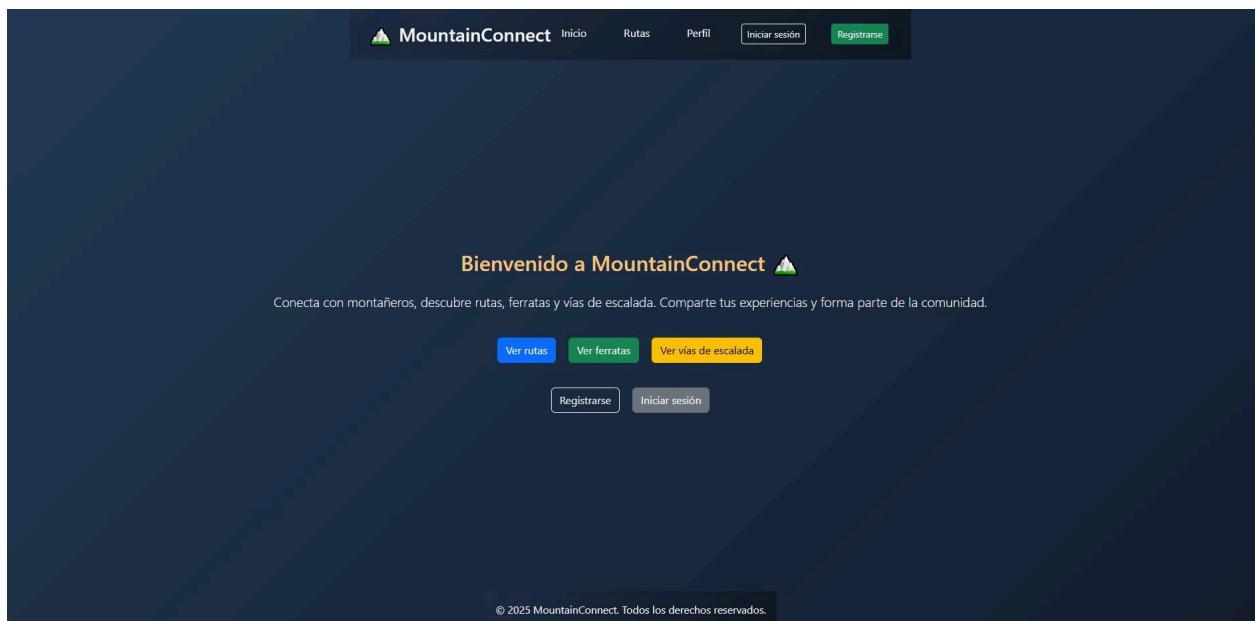
Página de inicio, iniciando sesión:



En la página de *index.php*, aparece en el header, la opción de inicio, que es donde nos situamos, las rutas, el perfil y cerrar sesión del usuario. En caso de que el usuario no esté registrado o no haya iniciado sesión, en vez de salir la opción de *Cerrar sesión* Sólo aparecerá la opción de **Iniciar sesión y Registrarse**

Al igual que al no haber iniciado sesión, en el cuerpo de la página aparecen las otras 2 opciones que no aparecen en la página de un usuario que ya tiene la sesión inicializada. También tiene la diferencia de que el usuario no registrado no le aparece el botón de *Crear ruta*

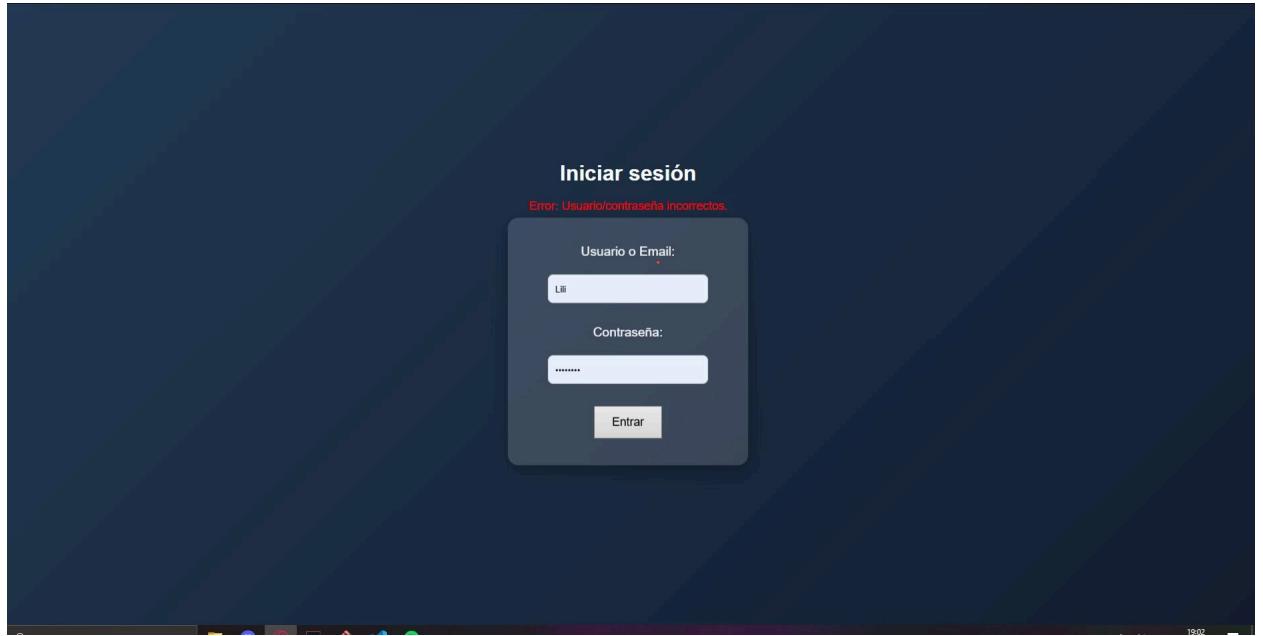
Página de inicio sin iniciar sesión:



Al cerrar sesión, se borra tu cuenta automáticamente porque

```
session_unset(); # Elimina todas las variables de sesión actuales.  
session_destroy(); # Destruye la sesión completa en el servidor.
```

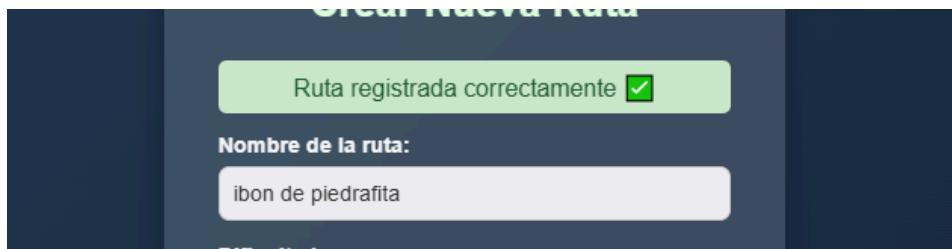
del archivo logout.php



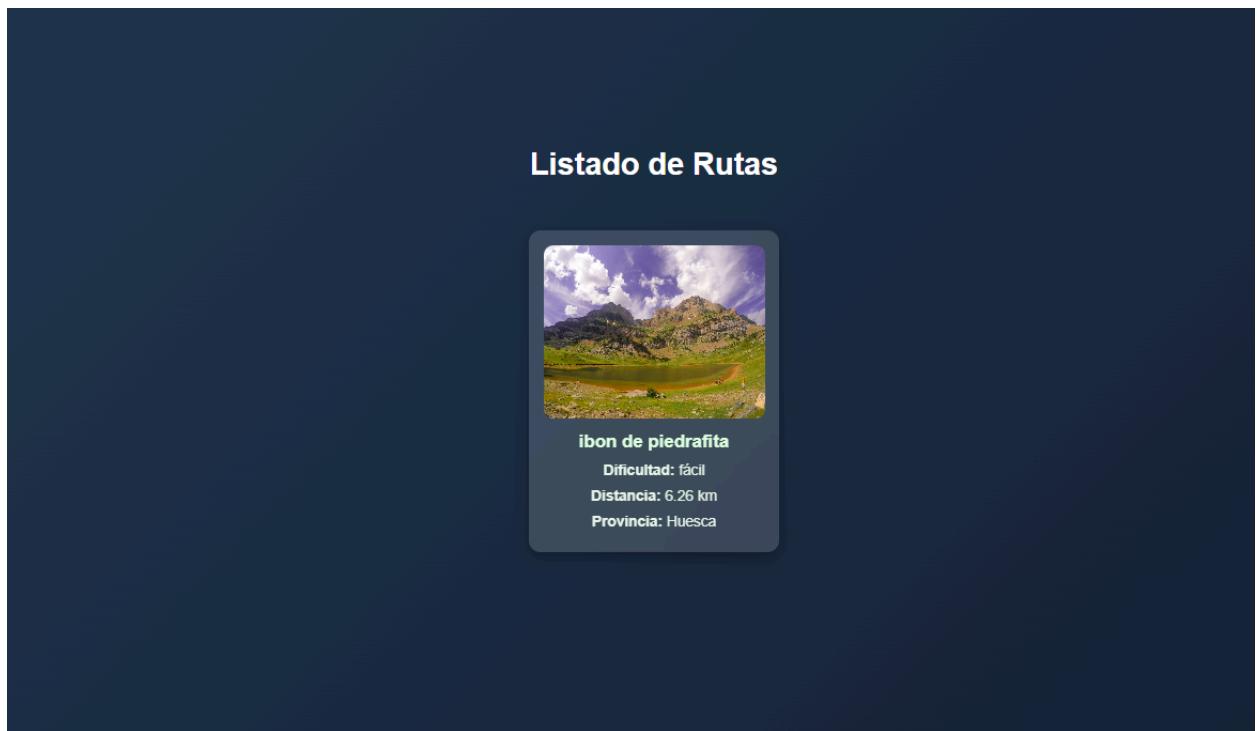
Al cerrar sesión el usuario deja de existir porque el archivo `logout.php` destruye la sesión.

Crear ruta `public/routes/create.php` introduce todos los datos de la ruta, añades imagen si quieres, cuando se añade la imagen, se manda directamente a la ruta `/uploads/photos` y se crea una imagen guardada llamada ruta\_numeros

Cuando guardas la ruta, para saber que lo has creado bien, te da la confirmación



Ahora, si vamos a `public/routes/list.php` podremos ver la ruta creada



## 4. Pruebas.

### Registro de usuario

Usuario ya creado Lili, [lili@gmail.com](mailto:lili@gmail.com), 12341234, 12341234, Escalada, Barcelona

Prueba	Entrada	SALIDA	Resultado
Registrar un usuario sin nombre usuario	a <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341234, Principiante, Escalada, Barcelona	El nombre de usuario es obligatorio.	NOK
Registrar un usuario	Lili, a <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341234, Principiante, Escalada, Barcelona	El nombre del usuario ya está en uso	NOK
Registrar un usuario	lili, lili <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341234, Principiante, Escalada, Barcelona	El email ya está registrado/ en uso	NOK
Registrar un usuario	lili, a <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341234, Principiante, Escalada, Barcelona	Error	OK
Contraseña sin poner	lili, a <a href="mailto:@gmail.com">@gmail.com</a> , Principiante ,Escalada, Barcelona	La contraseña es obligatoria. Debe confirmar la contraseña.	NOK
Contraseña>8dig	lili, a <a href="mailto:@gmail.com">@gmail.com</a> , 1234123, 1234123, Principiante, Escalada, Barcelona	La contraseña debe tener mínimo 8 caracteres.	NOK
confirmcontraseña!=Contraseña	lili, a <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341235, Principiante, Escalada, Barcelona	Las contraseñas no coinciden.	NOK

Correo	lili, <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341234, Principiante, Escalada, Barcelona	El email no tiene un formato válido.	NOK
Correo con mayuscula o minuscula	lili, LILI <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341234, Principiante, Escalada, Barcelona	El email ya está registrado / en uso	NOK
Correo vacío	lili, 12341234, 12341234, Principiante, Escalada, Barcelona	El email es obligatorio.	NOK
Especialidad	lili, a <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341234, Principiante, Barcelona	La especialidad es obligatoria.	NOK
Provincia sin seleccionar	lili, a <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341234, Principiante, Escalada	Seleccione una provincia.	NOK
Nivel exp sin seleccionar	lili, a <a href="mailto:@gmail.com">@gmail.com</a> , 12341234, 12341234, Escalada, Barcelona	Seleccione su nivel de experiencia.	NOK

## Login

Usuario creado existente: Ale, 12341234 / ale@gmail.com , 12341234

Prueba	Entrada	SALIDA	Resultado
loguearse con correo	ale@gmail.com , 12341234	OK	Bienvenido, Ale  Email: ale@gmail.com
loguearse con usuario	Ale , 12341234	OK	Bienvenido, Ale  Email: ale@gmail.com

loguearse con usuario mod mayus minus	ale , 12341234	NOK	Error: Usuario/contraseña incorrectos.
contraseña mal	Ale , 123412348	NOK	Error: Usuario/contraseña incorrectos.

## 5. Conclusión

He aprendido a trabajar con PHP enfocado al desarrollo web, en este proyecto ha sido más en concreto a la gestión de sesiones, la validación de datos introducidos del usuario con filter\_var, isset y empty, además de formularios. Cómo mantener la sesión iniciada con \$\_SESSION, tener orden de las carpetas para que todo esté a la vista y sea intuitivo y se vea organizado, teniendo una carpeta donde se le da lo estético (css, js, images), otra carpeta para lo que se puede reusar de código (header, footer, auth\_check)... Mejoraría el proyecto reemplazando el sistema de almacenamiento por una BBD real. Tener el código más organizado y mejor comentado, Mejorar el diseño visual, hacer que cerrar sesión no borre el usuario, sino, solo cierre sesión, mejorar los errores que aparecen en pantalla (que se vean mejor). Mejora de la página: Añadir una búsqueda con filtros, título... etc, seguir a otros usuarios de la web.

Dificultades: Error al guardar los usuarios en el array temporal, ya que guardaba valores incorrectos y no actualizaba bien el registro. Hubo problemas en el login, ya que solo me dejaba iniciar sesión con la primera cuenta creada, y los otros creados, no me dejaba acceder, esto me daba error porque no estaba leyendo bien el array de usuarios almacenados en sesión.