

PROYECTO PHP

Plataforma Web Montañera con PHP y MySQL

1. INTRODUCCIÓN DEL PROYECTO

1.1 Descripción General

Este proyecto didáctico tiene como objetivo el desarrollo completo de una aplicación web dinámica utilizando PHP y MySQL, aplicando de forma práctica los contenidos trabajados en clase. A lo largo del proyecto se implementarán funcionalidades esenciales de un sitio web moderno: registro y autenticación de usuarios, gestión de datos mediante operaciones CRUD, validación de formularios, subida de archivos e integración con bases de datos. Aunque la temática propuesta es MountainConnect, una red social para montañeros, el alumnado podrá adaptar la temática del proyecto (por ejemplo, rutas gastronómicas, clubes deportivos, eventos culturales, etc.) siempre que se respeten las funcionalidades y características técnicas establecidas. Al finalizar, se deberá entregar una memoria del proyecto, que incluirá la descripción técnica del desarrollo, capturas de pantalla, estructura de la base de datos y el enlace al repositorio de GitHub donde se encuentre el código fuente completo y documentado.

MountainConnect es una plataforma web social destinada a la comunidad montañera donde los usuarios pueden:

- Compartir y descubrir rutas de senderismo
- Publicar información sobre vías ferratas
- Documentar vías de escalada
- Subir fotografías de sus aventuras
- Interactuar con otros montañeros mediante comentarios y valoraciones
- Crear su perfil personalizado de montañero

1.2 Objetivos Generales de Aprendizaje

Al finalizar el proyecto, el alumnado será capaz de:

1. Desarrollar aplicaciones web dinámicas utilizando PHP
2. Diseñar e implementar bases de datos relacionales con MySQL
3. Aplicar principios de programación orientada a objetos en PHP
4. Gestionar sesiones y autenticación de usuarios
5. Implementar operaciones CRUD completas
6. Validar formularios y gestionar subida de archivos
7. Aplicar medidas de seguridad básicas (SQL injection, XSS, hash de contraseñas)
8. Estructurar código de forma mantenible y escalable

1.3 Tecnologías Utilizadas

- **Lenguaje Backend:** PHP 7.4 o superior
- **Base de datos:** MySQL 8.0 o MariaDB
- **Frontend:** HTML5, CSS3, Bootstrap
- **Servidor local:** XAMPP, WAMP o similar
- **Control de versiones:** Git necesario subir el codifo a vuestro repositorio.

1.4 Metodología de Trabajo

El proyecto se desarrollará de forma **incremental y progresiva**, dividido en tres unidades didácticas:

- **Unidad 3:** Fundamentos PHP y formularios básicos
- **Unidad 4:** Integración con bases de datos
- **Unidad 5:** Refactorización a Programación Orientada a Objetos

Cada unidad construye sobre la anterior, permitiendo ver la evolución natural del código desde un enfoque procedural hasta uno completamente orientado a objetos.

1.5 Resultados Esperados

Al finalizar el proyecto, se dispondrá de una aplicación web funcional con:

- Sistema completo de registro y autenticación
- Gestión de perfiles de usuario
- CRUD de rutas de senderismo
- CRUD de vías ferratas
- CRUD de vías de escalada
- Galería de fotografías asociadas a actividades
- Sistema de comentarios
- Sistema de valoraciones/likes
- Panel de administración básico

2. LENGUAJE DE PROGRAMACIÓN PHP

2.1 Objetivos Específicos de la Unidad

- Comprender la sintaxis básica de PHP
- Trabajar con variables, constantes y tipos de datos
- Implementar estructuras de control (condicionales, bucles)
- Crear y utilizar funciones
- Procesar formularios con métodos GET y POST
- Validar datos del lado del servidor
- Gestionar sesiones y cookies
- Implementar un sistema de login básico
- Manejar subida de archivos

2.2 Contenidos Teóricos

2.2.1 Fundamentos PHP

- Sintaxis básica y etiquetas PHP
- Variables y tipos de datos
- Operadores (aritméticos, lógicos, comparación)
- Arrays (indexados y asociativos)
- Constantes y variables superglobales (\$_GET, \$_POST, \$_SESSION, \$_FILES)

2.2.2 Estructuras de Control

- Condicionales: if, else, elseif, switch
- Bucles: for, foreach, while, do-while
- Include y require

2.2.3 Funciones

- Declaración y llamada de funciones
- Parámetros y valores de retorno
- Funciones útiles de PHP (string, array, date)

2.2.4 Formularios

- Métodos GET y POST
- Validación de datos
- Sanitización de entrada
- Mensajes de error y feedback
- Mantenimiento de valores en formularios

2.2.5 Sesiones

- Inicio y cierre de sesiones
- Almacenamiento de datos en sesión
- Destrucción de sesiones (logout)
- Control de acceso básico

2.2.6 Manejo de Archivos

- Subida de archivos (\$_FILES)
- Validación de tipo y tamaño
- Almacenamiento seguro
- Nombres únicos de archivo

2.3 Estructura del Proyecto - Fase 1

```
mountain-connect/
|
+-- public/
|   +-- index.php
|   +-- login.php
|   +-- register.php
|   +-- logout.php
|   +-- profile.php
|   |
|   +-- routes/
|       +-- list.php
|       +-- create.php
|       +-- view.php
|       +-- edit.php
|       +-- delete.php
|   |
|   +-- ferratas/
|   +-- climbing/
|       +-- photos/
|
+-- includes/
    +-- header.php
    +-- footer.php
    +-- auth_check.php
    +-- functions.php
|
+-- assets/
    +-- css/
    +-- js/
    +-- images/
```

```
|- uploads/
|   |- photos/
|   |- profiles/
|
└── config/
```

2.4 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 completa
- Archivos básicos creados
- Header y footer con HTML básico
- Archivo de configuración inicial

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

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

Diagrama de flujo de autenticación:

```
[ mermaid flowchart TD A[Usuario en login.php] --> B[Envía formulario] B --> C{¿Campos vacíos?} C -->|Sí| D[Mostrar errores] C -->|No| E{¿Credenciales válidas?} E -->|No|
```

```
F[Error: Usuario/contraseña incorrectos] E -->|Sí| G[Crear sesión] G --> H[Guardar user_id, username en $_SESSION] H --> I[Redirigir a index.php] D --> A F --> A ]
```

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

Tarea 1.5: Creación de rutas con galería de fotos asociada (8 horas)

ObjetivoDiseñar 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/

Tarea 1.6: Sistema de funciones auxiliares

Objetivo: Crear biblioteca de funciones reutilizables.

Funcionalidades:

- Función para sanitizar entrada de texto (opcional)
- Función para validar email
- Función para generar nombre único de archivo (opcional)
- Función para formatear fechas (opcional)
- Función para calcular dificultad en texto desde número (opcional)
- Función para verificar sesión activa

Archivo: includes/functions.php

2.5 Criterios de Evaluación - Entrega 1

Entregar una memoria donde se justifiquen los diseños de implementacion de vuestro proyecto y donde adjuntemos el enlace a vuestro repositorio en github.

Criterio	Puntos	Descripción
Estructura del proyecto	10	Organización correcta de carpetas y archivos
Formularios funcionales	15	Todos los formularios envían y procesan datos
Validación de datos	15	Validaciones completas del lado del servidor
Gestión de sesiones	15	Login/logout funcionan correctamente
Subida de archivos	10	Imágenes se suben y validan correctamente
Funciones auxiliares	10	Código reutilizable y bien organizado
Seguridad básica	10	Sanitización de entrada, verificación de sesiones
Código limpio	10	Indentación, comentarios, nombres descriptivos
Experiencia de usuario	5	Mensajes claros, navegación intuitiva
TOTAL	100	