

Documentación del Proyecto: Gestión de Mapas y Usuarios

Carlos Arturo Moreno Susatama

7 de noviembre de 2024

Introducción

Este documento detalla la estructura de archivos y la funcionalidad de cada uno en el proyecto de gestión de mapas y usuarios en un entorno orientado a objetos (POO) con PHP y MySQL. El objetivo es proporcionar una guía para entender cómo cada archivo contribuye al funcionamiento general del sistema.

Estructura de Archivos

La estructura de archivos del proyecto es la siguiente:

- **database.php** - Archivo de conexión a la base de datos.
- **user.php** - Clase para la gestión de usuarios.
- **map.php** - Clase para la gestión de mapas.
- **get_map.php** - Archivo para obtener el mapa de un usuario.
- **save_map.php** - Archivo para guardar el mapa de un usuario.
- **login.php** - Archivo para el inicio de sesión de los usuarios.
- **register.php** - Archivo para el registro de nuevos usuarios.
- **index.html** - Interfaz de usuario básica.

A continuación, se describe la funcionalidad de cada archivo.

Archivos y Funcionalidades

database.php

Este archivo contiene la clase **Database**, que es responsable de gestionar la conexión con la base de datos. La clase utiliza la extensión MySQLi de PHP para conectar, consultar, insertar, actualizar y eliminar datos de la base de datos.

Métodos principales:

- `connect()` - Método privado que establece la conexión con la base de datos.
- `query()` - Ejecuta una consulta SQL preparada.
- `close()` - Cierra la conexión de la base de datos.

user.php

Este archivo contiene la clase **User**, que permite la gestión de usuarios. Contiene métodos para registrar y autenticar a los usuarios en el sistema.

Métodos principales:

- `register()` - Registra un nuevo usuario en la base de datos.
- `login()` - Autentica a un usuario, estableciendo una sesión si las credenciales son correctas.

map.php

El archivo `map.php` define la clase **Map**, la cual permite la gestión de mapas en el sistema. La clase contiene métodos para guardar y recuperar el mapa de un usuario específico.

Métodos principales:

- `saveMap()` - Guarda los datos del mapa de un usuario en la base de datos.
- `getMap()` - Recupera el mapa de un usuario desde la base de datos.

get_map.php

Este archivo permite obtener el mapa de un usuario autenticado. Valida la sesión del usuario y utiliza la clase **Map** para recuperar el mapa correspondiente de la base de datos.

save_map.php

Este archivo permite guardar el mapa de un usuario autenticado. Valida la sesión del usuario y utiliza la clase **Map** para almacenar los datos del mapa en la base de datos.

login.php

El archivo `login.php` procesa el inicio de sesión del usuario. Recibe las credenciales del usuario, valida su autenticidad mediante la clase `User` y establece una sesión en caso de éxito.

register.php

El archivo `register.php` permite registrar a nuevos usuarios. Recibe los datos del usuario y llama al método `register()` de la clase `User` para almacenar al usuario en la base de datos.

index.html

Este archivo es una interfaz de usuario básica que permite a los usuarios registrarse, iniciar sesión, guardar su mapa y recuperarlo. La interfaz utiliza JavaScript para realizar peticiones `fetch()` a los archivos PHP correspondientes.

Diagrama de Interacciones

A continuación se presenta una descripción del flujo de interacción entre los archivos:

1. `index.html` envía solicitudes de registro y de inicio de sesión a `register.php` y `login.php`, respectivamente.
2. Tras iniciar sesión, `index.html` permite al usuario guardar el mapa mediante `save_map.php` o recuperarlo mediante `get_map.php`.
3. Los archivos `save_map.php` y `get_map.php` interactúan con la clase `Map` para realizar operaciones en la base de datos.

Estructura de Archivos en MVC

La siguiente estructura muestra el diseño del proyecto basado en el patrón Modelo-Vista-Controlador (MVC), indicando la distribución de las carpetas y los archivos:

```
project-root
├── config
│   └── database.php
├── public
│   └── index.php
```

login.php
register.php
save_map.php
get_map.php

Models

User.php
Map.php

Views

login_view.php
register_view.php
map_view.php
home.php

Controllers

UserController.php
MapController.php

