Actualización del Enunciado: Desarrollo de un CRUD con PHP, Docker, Composer y DotEnv

Título: Construcción de un Sistema CRUD con Registro y Login usando PHP, MySQL y Docker

Objetivo General:

Implementar un sistema CRUD (Crear, Leer, Actualizar, Borrar) para gestionar los datos de una tienda. Este sistema incluirá funcionalidad de **registro y login**, garantizando que en todo momento un usuario debe estar registrado para interactuar con la aplicación. El sistema deberá informar mediante mensajes las acciones realizadas y el usuario conectado deberá mostrarse en todo momento.

Detalles del Proyecto:

Funcionalidades Principales:

1. Registro y Login en index.php:

- Registro:
 - Permite crear un nuevo usuario con un nombre único y contraseña.
 - La contraseña debe almacenarse de forma segura utilizando password_hash().
- Login:
 - Valida las credenciales del usuario contra los datos almacenados en la base de datos.
 - Si el usuario es autenticado correctamente, se almacena su información en la sesión.
- En ambos casos, muestra mensajes claros al usuario sobre el éxito o fallo de la acción realizada.

2. Persistencia del Usuario:

- En todo momento, si un usuario está registrado, debe mostrarse su nombre en la aplicación.
- Permitir al usuario desconectarse con un botón Logout.

3. Restricción de Acceso:

- Las páginas internas (Sitio.php, listado.php, etc.) deben estar protegidas.
- Si no hay un usuario en sesión, redirigir automáticamente a index.php.

4. **CRUD**:

- Crear, Leer, y Borrar registros en las tablas de la base de datos. Opcionalmente se podrá modificar datos, pero no se pide dicha funcionalidad
 - productos
 - tiendas
 - usuarios
 - stock

- familia
- Muestra un mensaje cada vez que se realiza una acción (p. ej., "Producto añadido correctamente", "Error al eliminar producto, puede ser que haya integridad referencial").

Requisitos Técnicos:

1. Base de Datos:

- Usa la base de datos tienda configurada en el contenedor MySQL.
- El archivo datos. sql inicializa las tablas con datos de ejemplo.

2. Docker:

- Usa el archivo docker-compose.yaml proporcionado para levantar los contenedores:
 - web (PHP + Apache).
 - mysql (base de datos).
 - phpmyadmin (gestión visual de la base de datos).

3. Composer y DotEnv:

- Usa Composer para instalar dependencias.
- Usa la librería vlucas/phpdotenv para manejar los parámetros de conexión a la base de datos.
- Los parámetros como DB_HOST, DB_USER, DB_PASS, y DB_NAME deben configurarse en un archivo . env.

4. Sesiones:

• Utiliza sesiones PHP para manejar la autenticación del usuario y su estado en la aplicación.

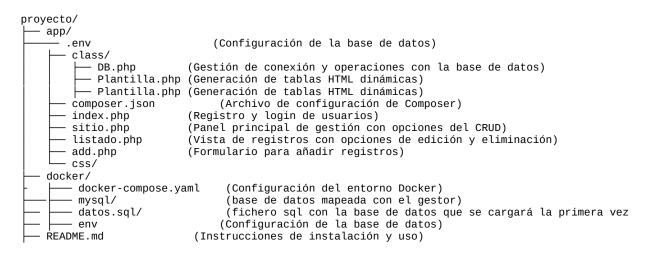
5. Mensajes Informativos:

- Implementa un sistema para mostrar mensajes en cada acción del usuario.
- Ejemplos:
 - "Usuario registrado con éxito."
 - "Credenciales incorrectas."
 - "Producto eliminado correctamente."

6. Uso de PDO o MySQLi:

- Se ha explicado en la clase, con mysqli, pero se puede user**PDO** por su flexibilidad, soporte para múltiples bases de datos y su manejo integrado de excepciones.
- Implementa consultas con sentencias preparadas para evitar invecciones SQL.

Estructura del Proyecto Actualizada:



Rúbrica de Corrección Actualizada

Criterio Descripción

Configuración Docker El entorno Docker está configurado y funcional.

Uso de Composer y Se usa Composer para instalar dependencias y DotEnv para

DotEnv manejar la configuración.

Autenticación con Login, registro, logout y persistencia del usuario funcionan

Sesiones correctamente.

Mensajes Se muestran mensajes claros al usuario en todas las

Informativos acciones realizadas.

Todas las operaciones CRUD funcionan correctamente en Funcionalidad CRUD

todas las tablas.

Se utiliza PDO para todas las operaciones con la base de Uso de mysqli o PDO

datos con sentencias preparadas.

El código está organizado en clases y carpetas, siguiendo la

estructura propuesta.

Estructura del **Proyecto**

Uso de clases correctas Implementación de los

métodos

Claridad y legibilidad

Navegación de la aplicación

Los valores rojos representan posibles variables de sesión

Los valores negros representan valores de formularios

