Práctica 9 – Acceso a Datos con PDO y Transacciones

Objetivo:

Diseñar un pequeño sistema en PHP que gestione mediante PDO y transacciones las actividades relacionadas con tu hobby personal (por ejemplo, Lectura, Ciclismo, Cocina, Senderismo, etc.).

Crea el proyecto nuevo con composer, crea carpeta "public" para los archivos públicos y crea el directorio **"src"** para almacenar las clases de .php y otros scripts. En el directorio **"tools"** dejaras aquellas clases que te sirvan como herramienta para otros proyectos.

Para poder entrar a tu nuevo gestor de hobbies debe ser un usuario autenticado. (login contra bbdd)

1️⃣ Creación de la Base de Datos

Crea una Base de Datos en MySQL que contenga **una tabla que represente las actividades o registros asociados a tu hobby personal**.

Por ejemplo, si tu hobby es *Lectura*, la tabla podría llamarse `lectura` y contener información sobre los libros leídos.

Instrucciones:

1.1) La tabla tendrá tantas columnas como atributos describan una actividad de tu hobby.
Ejemplo para *Lectura*:

- `id` (PK autoincremental)
- `titulo_libro` (cadena, UNIQUE)
- `autor` (cadena)
- `paginas` (entero)
- `terminado` (booleano)
- `fecha_lectura` (DATE)

1.2) Define una **clave primaria autoincremental (PK)** y uno o varios campos **UNIQUE** (por ejemplo, el título).

1.3) Incluye al menos un atributo de cada tipo de dato primitivo:

- Entero (INT)
- Booleano (TINYINT(1))

- Cadena (VARCHAR)
- Fecha (DATE)

2️⃣ Clase de conexión (PDO)

Crea una clase denominada `Conexion` con un método estático `getConexion()` que devuelva un objeto **PDO**.
Configura PDO con:

- Juego de caracteres `utf8mb4`
- Control de errores mediante excepciones (`PDO::ERRMODE_EXCEPTION`).

3️⃣ Clase Gestor de tu Hobby

Crea una clase denominada `Gestor<MiHobby>` (por ejemplo, `GestorLectura`) que implemente una interfaz `AccionesBD` con los métodos básicos CRUD:

Método	Descripción
insertar(array $datos)	Inserta una nueva actividad de tu hobby
eliminar(int $id)	Elimina una actividad por su PK
actualizar(int $id, array $datos)	Actualiza los campos de una actividad
listar(): array	Devuelve todas las actividades en un array de objetos

Requisitos:

- Maneja excepciones tanto de conexión como de SQL.
- Devuelve o lanza mensajes de error claros.

4️⃣ Script de visualización

Crea un archivo `tabla_<mihobby>.php` (por ejemplo, `tabla_lectura.php`) que muestre todos los registros en una tabla HTML.
Debe incluir en cada fila:

- Un enlace para **borrar** el registro.
- Un enlace para **editar** el registro.

Utiliza una instancia de tu clase `Gestor<MiHobby>` para mostrar los datos.

5 Transacciones – Altas Simultáneas

Agrega en `Gestor<MiHobby>` un método **private** denominado `altaSimultanea(array $registros)` que reciba **varios arrays asociativos**, cada uno representando una nueva actividad de tu hobby.

El método debe:

1. Iniciar una **transacción** (`beginTransaction()`),
2. Preparar una **sentencia INSERT** con parámetros (`prepare()`),
3. **Iterar** los registros e ir ejecutando los `execute()`,
4. Hacer `commit()` si todas las inserciones son correctas,
5. Si alguna inserción provoca error (por ejemplo, un valor duplicado en un campo UNIQUE), capturar la excepción y realizar un **rollback()**, de forma que **no se inserte ningún registro**.

Pruebas obligatorias:

- ✅ **Prueba A:** Invoca el método con varios registros válidos → Se insertan todos.
- ❌ **Prueba B:** Invoca el método con dos registros que tengan el mismo valor en un campo UNIQUE (por ejemplo, mismo título) → Se cancela la transacción y no se inserta ninguno.

Notas técnicas:

- La tabla debe ser **InnoDB**.
- El modo de error de PDO debe ser `PDO::ERRMODE_EXCEPTION`.
- *No se debe indicar la PK en el `INSERT` (es autoincremental).
- Pueden añadirse métodos de test para verificar el correcto funcionamiento.

6 Evaluación

Criterio	Ponderación
Creación correcta de la BD y tipos de datos	15%
Clase de conexión con PDO y excepciones	15%
CRUD completo y funcional	30%

Transacciones correctamente implementadas	25%
Calidad del código, validaciones y legibilidad	15%