

PRÁCTICA 1 UD7

TRIMESTRE: SEGUNDO

Fecha:

CICLO: Desarrollo de Aplicaciones Web

CURSO: 2º

CALIFICACIÓN:

MÓDULO: Desarrollo Web Entorno Servidor

Turno: Mañana

Nombre:

Apellidos:

Instrucciones: Esta práctica tiene como finalidad asentar los conocimientos de Acceso a Datos con PHP (PDO). Lee atentamente y responde desarrollando el código en un archivo llamado `install_pettinder.php`.

PARTE PRÁCTICA. TIPO A.

- Esta práctica debe entregarse subiendo el archivo `.php` a la plataforma.
- El código debe ser limpio, comentado y funcional.

ESCENARIO: PetTinder (Start-up de Adopción)

Has sido fichado como Backend Developer para la nueva app de moda: "PetTinder". La aplicación permitirá a los usuarios dar "Match" a mascotas que buscan hogar.

Tu misión hoy es programar el Script de Instalación (install.php). Este script se ejecutará cada vez que queramos "resetear" el servidor, por lo que debe ser robusto y automático.

EJERCICIO 1: Conexión e Idempotencia (Configuración)

Debes establecer la conexión inicial con el servidor de base de datos.

- Conéctate al servidor MySQL local (localhost) usando el usuario root.
- Configura el manejo de errores para que PHP lance Excepciones (ERRMODE_EXCEPTION) si algo falla.
- Requisito de Idempotencia: El script debe poder ejecutarse 100 veces sin dar error. Para ello, programa una instrucción SQL que elimine la base de datos pettinder_db solo si ya existe.
- Crea la base de datos pettinder_db obligando al uso de caracteres utf8mb4 (para soportar emojis en las biografías de los animales) y selecciónala para empezar a trabajar.

EJERCICIO 2: Arquitectura de Datos (DDL)

Diseña la tabla principal donde se guardarán los perfiles. Crea la tabla mascotas con las siguientes columnas:

- id: Numérico, Clave Primaria y Auto-incremental.
- nombre: Texto (máximo 50 caracteres), obligatorio.
- bio: Texto largo para la descripción de la mascota.
- likes: Numérico entero, valor por defecto 0.
- Importante: Define explícitamente el motor InnoDB al final de la creación de la tabla para soportar futuras transacciones.

EJERCICIO 3: Seeding y Seguridad (DML)

La base de datos no puede nacer vacía. Debes insertar datos de prueba ("Semillas").

- Inserta al menos 2 mascotas (Ej: "Firulais" y "Michi").
- Seguridad: Está terminantemente prohibido concatenar variables en la cadena SQL. Debes utilizar Sentencias Preparadas (prepare) y vinculación de parámetros (execute con array o bindParam) para realizar estas inserciones.

Extras para sumar nota!!!

EJERCICIO 4: Ingeniería de Datos Avanzada (Relaciones N:M)

Modifica y amplía tu archivo install_pettinder.php para que, además de la tabla mascotas, cree la infraestructura para usuarios y likes.

- Crea la tabla usuarios con: id (PK), email (Debe ser UNIQUE) y password (VARCHAR 255).
- Crea la tabla matches (Tabla pivote) que relacione a un usuario con una mascota.
- Restricción de Integridad: Un usuario no puede dar like dos veces a la misma mascota (Debes configurar una Clave Primaria Compuesta o un Índice Único).
- Integridad Referencial (FK):
 - Si se borra una mascota, sus likes deben desaparecer automáticamente (ON DELETE CASCADE).
 - Si un usuario intenta borrarse, pero tiene likes activos, la base de datos debe impedirlo (ON DELETE RESTRICT).

EJERCICIO 5: Transacciones ACID (Lógica de Negocio)

Vamos a implementar la funcionalidad de "Dar Like" de forma segura. Crea un archivo llamado `match_process.php`.

- Simula en el código que el Usuario 1 da like a la Mascota 1.
- Debes implementar una Transacción ACID ("Todo o Nada") que realice dos operaciones simultáneas:
 1. Insertar el registro en la tabla matches.
 2. Incrementar en +1 el contador likes de la tabla mascotas (usando UPDATE).
- Controla la transacción: Si algo falla (ej: el usuario ya dio like y salta la restricción del Ejercicio 4), haz un Rollback. Si todo funciona, haz Commit.

EJERCICIO 6: Consultas Inteligentes (Mapeo de Objetos)

Queremos dejar de usar arrays asociativos simples.

- Define una clase en PHP llamada Mascota cuyas propiedades públicas coincidan con las columnas de tu base de datos.
- Añade un método lógico a la clase: `public function getEstado()`.
 - Si la mascota tiene más de 10 likes, devuelve: "POPULAR".
 - Si tiene 10 o menos, devuelve: "Nuevo".
- Realiza una consulta SELECT utilizando `PDO::FETCH_CLASS` para obtener instancias de esta clase automáticamente y muestra el nombre y el estado de cada mascota en una lista HTML.

FASE 1: Instalación e Infraestructura

[OK] Base de datos antigua eliminada (Limpieza).

[OK] Base de datos 'pettinder_db' creada y seleccionada.

[OK] Tablas 'usuarios' y 'mascotas' creadas (InnoDB).

[OK] Tabla 'matches' creada con Claves Foráneas.

[OK] Datos insertados correctamente mediante sentencias preparadas.

FASE 2: Transacciones y Objetos (Extras)

[TRANSACCIÓN EXITOSA] Se ha registrado el Match y actualizado el contador.

Listado de Mascotas (Modo Objeto)

Firulais (Likes: 6) **NUEVO**

Michi (Likes: 15) **POPULAR**