

PRÁCTICO 3

Base de Datos

1. Piense/busque ejemplos concretos de porque se utilizan archivos en vez de una BBDD.
2. Responda y justifique
 - a. Las imágenes y videos, se pueden almacenar en una BBDD ?
 - b. ¿Cómo piensa que se puede tener asociada una imagen a un registro (por ejemplo, la foto carnet de un usuario?)

EJERCICIOS

3. Cree una base de datos para registrar pagos de deudas. Directamente desde PhpMyAdmin o cualquier cliente gráfico, sin usar PHP.
 - a. Cree la tabla **pagos** con la siguiente información

deudor: Nombre de la persona
cuota : Número de cuota pagada
cuota_capital: Monto de la cuota pagada
fecha_pago: Fecha que se realizó el pago de la cuota

¿Es necesario agregar una clave primaria?
 - b. Inserte varios **registros** “a mano”.
 - c. Realize consultas SQL para:
 - i. Listar todos los pagos
 - ii. Listar sólo aquellos pagos que pertenezcan a un deudor determinado
 - iii. Listar los pagos mayores a un monto determinado
 - iv. Eliminar un pago determinado.
 - v. Insertar un pago
 - vi. Actualizar la cuota_capital de un pago determinado
 - vii. Obtener el promedio de las cuotas pagadas
 - viii. Obtener la cantidad de cuotas pagadas por un deudor determinado

4. Usando la tabla **pagos** de la base de datos del punto 3, desde PHP:

- a. Liste todos los registros (filas) de la tabla en una tabla HTML.
- b. Ingrese algunos valores de ejemplo directamente desde un script php (sin formulario)
- c. Cree un formulario para ingresar pagos a la base de datos.
- d. Como podria hacer para evitar que el sistema no permita registrar dos veces el mismo pago?

NOTA: Primero pruebe las consultas SQL desde phpMyAdmin (o cualquier cliente SQL) y una vez listas pasarlas a PHP+PDO.

5. Un sistema web universitario debe registrar la información de sus materias. Dispone de la siguiente tabla:

MATERIA(id: int, nombre: string; profesor: string;)

- a. Crear la base de datos para soportar el modelo de datos planteado.
- b. Añada una interfaz web para realizar el **ABM*** completo de materias.
- c. Realice un buscador de materias por nombre. Investigue el operador de comparación [LIKE](#) de SQL.

*El término **ABM** (Alta, Baja y Modificación) o **CRUD** (del inglés Create, Read, Update, and Delete) se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.

6. Modifique el ejercicio 5 para agregar la **CARRERA** a la que pertenece cada materia. Las carreras además del nombre, deben guardar la duración en años.

- ¿Qué modificaciones en la base de datos tengo que hacer? ¿Cómo se relacionan las carreras y las materias?

Incluya los siguientes servicios en el sistema:

- i. Listar las materias de una carrera seleccionada
- ii. Listar las carreras con duración menor a 3 años.