PRÁCTICO 3  
Base de Datos

1. Piense/busque ejemplos concretos de porque se utilizan archivos en vez de una BBDD.

* **Mejor rendimiento,** si almacena archivos grandes en bases de datos, puede ralentizar el rendimiento porque una consulta simple para recuperar la lista de archivos o el nombre de archivo también cargará los datos del archivo si utilizó Select \* en su consulta. Si bien el sistema de Archivos que accede a un archivo es bastante simple y ligero.
* **Simplicidad en su almacenamiento**, guardar los archivos y descargarlos en el sistema de archivos es mucho más simple que la base de datos, ya que una simple función Guardar como le ayudará. La descarga se puede hacer dirigiendo una **URL** con la ubicación del archivo guardado.

1. Responda y justifique
   1. Las imágenes y videos, se pueden almacenar en una BBDD ?
   2. ¿Cómo piensa que se puede tener asociada una imagen a un registro (por ejemplo, la foto carnet de un usuario?)

**EJERCICIOS**

1. Cree una base de datos para registrar pagos de deudas. Directamente desde PhpMyAdmin o cualquier cliente gráfico, sin usar PHP.
   1. 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?

* 1. Inserte varios **registros** “a mano”.
  2. Realize consultas SQL para:
     1. Listar todos los pagos

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `pagos` me trae toda la lista

* + 1. Listar sólo aquellos pagos que pertenezcan a un deudor determinado

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `pagos` WHERE deudor='Luciano Arias'

La cláusula where se utiliza en los casos que no requerimos que devuelvan todos los registros de una tabla, sino que unicamente los datos que cumplan ciertas condiciones.

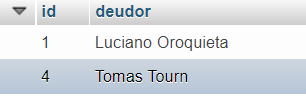
* + 1. Listar los pagos mayores a un monto determinado

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `pagos` WHERE cuota\_capital > 3000



* + 1. Eliminar un pago determinado.

[DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/delete.html) FROM `pagos` WHERE id=2



* + 1. Insertar un pago



* + 1. Actualizar la cuota\_capital de un pago determinado



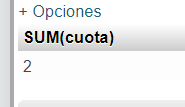
* + 1. Obtener el promedio de las cuotas pagadas

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [AVG](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/group-by-functions.html#function_avg)(cuota\_capital) FROM `pagos`



* + 1. Obtener la cantidad de cuotas pagadas por un deudor determinado

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/group-by-functions.html#function_sum)(cuota) FROM `pagos` WHERE deudor='Tomas Tourn'



1. Usando la tabla **pagos** de la base de datos del punto 3, desde PHP:
   1. Liste todos los registros (filas) de la tabla en una tabla HTML.
   2. Ingrese algunos valores de ejemplo directamente desde un script php (sin formulario)
   3. Cree un formulario para ingresar pagos a la base de datos.
   4. 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.

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

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

* 1. Crear la base de datos para soportar el modelo de datos planteado.
  2. Añada una interfaz web para realizar el **ABM\*** completo de materias.
  3. Realice un buscador de materias por nombre. Investigue el operador de comparación [LIKE](https://dev.mysql.com/doc/refman/8.0/en/string-comparison-functions.html) 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.

1. 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:

* + 1. Listar las materias de una carrera seleccionada
    2. Listar las carreras con duración menor a 3 años.