

Esta clase va a ser

- grabada

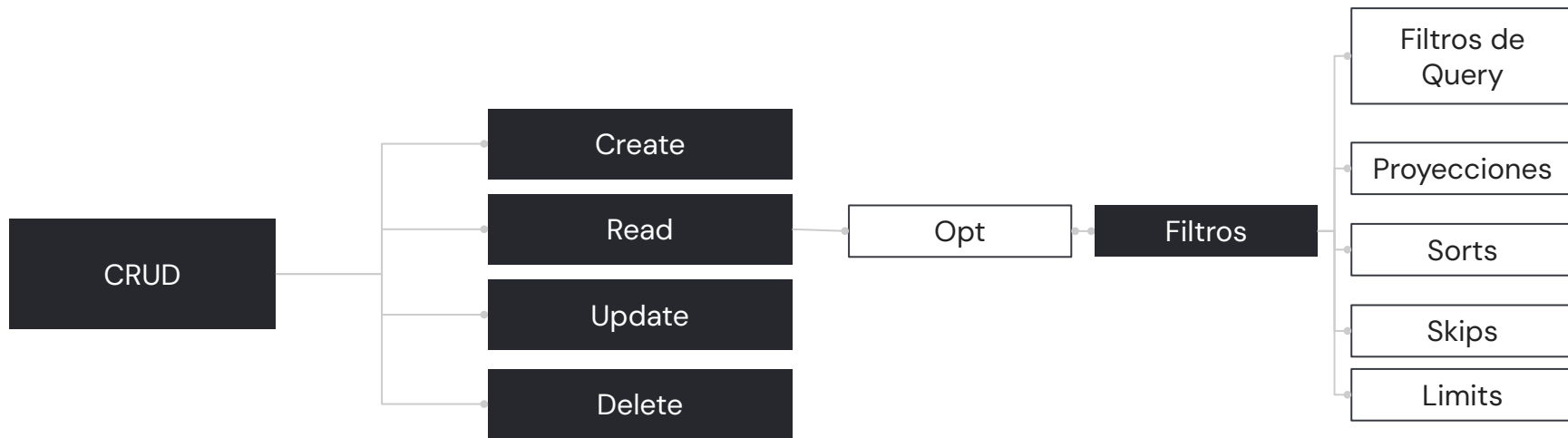
**Clase 13.** DESARROLLO AVANZADO DE BACKEND

# CRUD en MongoDB

# Objetivos de la clase

- Comprender el concepto de CRUD
- Aplicar el CRUD a la base de datos de MongoDB

## MAPA DE CONCEPTOS



# CRUD

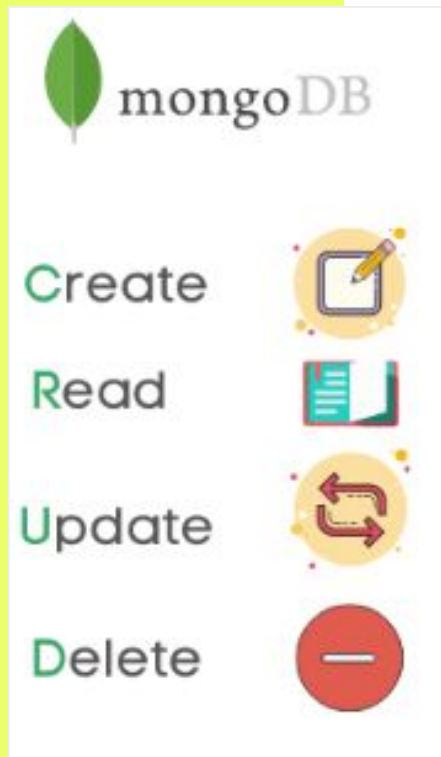
CRUD es un acrónimo que hace referencia a las cuatro operaciones fundamentales de una base de datos:

- ✓ C : Create (Crear un dato, insertarlo en la base de datos)
- ✓ R : Read (Leer un dato, mostrarlo al cliente)
- ✓ U : Update (Actualizar un dato, cambiar su información interna)
- ✓ D : Delete (Eliminar un dato, removerlo de nuestra colección)



# Primeros comandos

## CRUD: CR



- ✓ **`db.collection.insertOne(doc)`** : Agrega un nuevo documento a la colección seleccionada.
- ✓ **`db.collection.insertMany(docs)`**: Agrega múltiples documentos a la colección seleccionada (dado un arreglo de documentos).
- ✓ **`db.collection.findOne(opt)`**: Busca un elemento que cumpla con los criterios de búsqueda (opt), devuelve el primer documento que cumpla con dicho criterio.
- ✓ **`db.collection.find(opt)`**: Devuelve todos los documentos que cumplan con dicho criterio.
- ✓ **`db.collection.find(opt).pretty()`**: Añadido para hacer más presentables los resultados de un `find()`.

# Conteo de datos

## MongoDB count



Los comandos de conteo para determinar el número de documentos en una colección son:

- ✓ **`db.collection.estimatedDocumentCount()`** Cuenta el estimado más próximo al número de documentos según su metadata.
- ✓ **`db.collection.countDocuments(opt)`** Cuenta los documentos que cumplan con el criterio definido en las opciones (opt).



## Ejemplo en vivo

Se creará una base de datos llamada “baseCRUD”.

- ✓ Se agregará una colección llamada “mascotas”
- ✓ Se agregarán 3 mascotas con las propiedades: nombre, especie, edad
- ✓ Se buscarán mascotas por su especie
- ✓ Contar el número de mascotas totales agregadas.

Tiempo estimado: **10 minutos**





## ACTIVIDAD EN CLASE

# CRUD – CR

Sobre una base de datos llamada “colegio”, crear una colección “estudiantes” donde se agregarán documentos con los siguientes datos:

- ✓ nombre
- ✓ apellido
- ✓ curso
- ✓ edad
- ✓ correo
- ✓ sexo

Crear 5 estudiantes (Insert Many) con los campos mencionados arriba. Además, crear un estudiante sólo con nombre, apellido y curso. ¿Es posible?

- ✓ Realizar una búsqueda para obtener a todos los estudiantes.
- ✓ Realizar una búsqueda para obtener a todos los estudiantes de sexo H (hombre)
- ✓ Realizar un conteo para obtener el número de documentos totales.
- ✓ Realizar un conteo para obtener el número de documentos totales que cumplan con el criterio: “Es mujer”

# Búsquedas más complejas: filtros

# Filtros

Las búsquedas del mundo real no siempre requieren que un valor sea **igual** a otro. En ocasiones necesitamos que sea menor, mayor, diferente de, entre otras cosas.

Los filtros pueden agregarse dentro de los elementos de criterio (opt) con ayuda del símbolo **\$**, además, podemos agregar más de un filtro para asegurarnos que el documento se ajuste a criterios muy específicos.

Entonces, la sintaxis general será:

```
db.coll.find( {key: {$operator: val}} )
```



# MongoDB: Operadores para Filtros de Query

- ✓ **\$and** : Realiza operación AND -> sintaxis: `{ $and: [ {}, {} ] }`
- ✓ **\$or** : Realiza operación OR -> sintaxis: `{ $or: [ {}, {} ] }`
- ✓ **\$lt** : Coincide con valores que son menores que un valor especificado.
- ✓ **\$lte** : Coincide con valores menores o iguales a un valor especificado.
- ✓ **\$gt** : Coincide con valores mayores a un valor especificado.
- ✓ **\$gte** : Coincide con valores mayores o iguales a un valor especificado.
- ✓ **\$ne** : Coincide con valores que no son iguales a un valor especificado.
- ✓ **\$eq** : Selecciona los documentos que son iguales a un valor especificado.

# MongoDB: Operadores para Filtros de Query

- ✓ **\$exists** : Selecciona los documentos según la existencia de un campo.
- ✓ **\$in** : Selecciona los documentos especificados en un array.  
sintaxis: {key:{\$in: [array of values] } }
- ✓ **\$nin** : Coincide con ninguno de los valores especificados en un array.
- ✓ **\$size** : Coincide con el número de elementos especificados.
- ✓ **\$all** : Coincide con todos los valores definidos dentro de un array.
- ✓ **\$elemMatch** : Coincide con algún valor definido dentro del query.



# Break

¡10 minutos y volvemos!

# Proyecciones, Sorts, Skips y Limits



## Ejemplo en vivo

Basado en nuestra base de datos “colegio”

- ✓ Se agregarán 5 estudiantes más, con diferentes campos y con la misma estructura. además, crear 1 alumno sólo con nombre.
- ✓ Realizar una búsqueda aplicando ordenamientos, proyecciones, saltos y límites.
- ✓ Se analizarán los resultados de las proyecciones, saltos, ordenamientos y límites. ¿Cómo se comportan los documentos que tienen campos incompletos?



# CRUD – UD

## Update y Delete



# Operaciones con Filtros

Sobre la base y los datos cargados anteriormente

1. Insertar cinco documentos más en la colección clientes con los siguientes datos:  

```
{ "nombre": "Pablo", "edad": 25 }  
{ "nombre": "Juan", "edad": 22 }  
{ "nombre": "Lucia", "edad": 25 }  
{ "nombre": "Juan", "edad": 29 }  
{ "nombre": "Fede", "edad": 35 }
```
2. Listar todos los documentos de la colección clientes ordenados por edad descendente.
3. Listar el cliente más joven.
4. Listar el segundo cliente más joven.
5. Listar los clientes llamados 'Juan'
6. Listar los clientes llamados 'Juan' que tengan 29 años.
7. Listar los clientes llamados 'Juan' ó 'Lucia'.



## ACTIVIDAD EN CLASE

# Operaciones con Filtros

8. Listar los clientes que tengan más de 25 años.
9. Listar los clientes que tengan 25 años ó menos.
10. Listar los clientes que NO tengan 25 años.
11. Listar los clientes que estén entre los 26 y 35 años.
12. Actualizar la edad de Fede a 36 años, listando y verificando que no aparezca en el último listado.
13. Actualizar todas las edades de 25 años a 26 años, listando y verificando que aparezcan en el último listado.
14. Borrar los clientes que se llamen 'Juan' y listar verificando el resultado.
15. Eliminar además todos los documentos de estudiantes que hayan quedado con algún valor.

¿Preguntas?

**Opina y valora**  
esta clase

**Muchas gracias.**