

Operaciones CRUD

Las operaciones CRUD crean , leen , actualizan y eliminan documentos.

Crear operaciones

Las operaciones de creación o inserción agregan nuevos documentos a una colección. Si la colección no existe, las operaciones de creación también crean la colección.

Puede insertar un solo documento o varios documentos en una sola operación.

Insert() e insertMany()

Leer operaciones

Las operaciones de lectura recuperan documentos de una colección; es decir, consultar una colección de documentos.

Puede especificar criterios o filtros que identifiquen los documentos a devolver.

Find()

Actualizar operaciones

Las operaciones de actualización modifican los documentos existentes en una colección. Puede actualizar un solo documento o varios documentos en una sola operación.

Puede especificar criterios o filtros que identifiquen los documentos a actualizar. Estos filtros utilizan la misma sintaxis que las operaciones de lectura.

Update() - replaceOne()

Eliminar operaciones

Las operaciones de eliminación eliminan documentos existentes de una colección. Puede eliminar un solo documento o varios documentos en una sola operación.

Puede especificar criterios o filtros que identifiquen los documentos que se eliminarán. Estos filtros utilizan la misma sintaxis que las operaciones de lectura.

deleteOne() - deleteMany()

```
/* Las funciones para crear, consultar, actualizar y eliminar tienen la siguiente sintaxis */
```

```
Nombre_basededatos.nombre_coleccion.funcionCRUD();
```

```
/* MongoDB tiene funciones que permiten realizar las funciones CRUD de forma masiva o unica */
```

```
# CREAR BASE DE DATOS !!
```

```
# para crear una base de datos se usa en la terminal el comando mongo para iniciar
```

```
# para crear base de datos
```

```
use nombre_db
```

```
# para crear mi primera colección uso
```

```
db.movie.insert({"Doc1":"Avengers"})
```

```
# para ver las bases de datos que tenemos podemos usar
```

```
show dbs
```

```
show collections
```

```
# para crear multiples documntos comando insertMany()
```

```
db.movie.insertMany([{"pelicula":"Scary movie", "año": 1990}, {"pelicula":"Rapido y furioso", "año": 2000}])
```

```
# Metodo busqueda find
```

```
db.movie.find() # regresa todos los documentos
```

```
db.movie.find({"pelicula":"Scary movie"}) # busca el o los documentos con el valor de scary movie
```

```
db.movie.find({"pelicula":"Scary movie"}, {"_id": 0}) # regresa el documento de scary movie pero sin el id
```

```
.find({"lo que busco"}, {"lo que no quiero que se vea"});
```

```
# Metodo replaceOne({},{}) reemplaza una caracteristica
```

```
db.movie.replaceOne({"nombre":"Avengers"}, {"pelicula":"Avengers", "año": 2010})
```

```

# upsert true busca un match y si no lo hay crea un documento con esa
informacion
db.movie.replaceOne({"nombre":"Avengers"}, {"pelicula":"Avengers", "año":
2010}, {upsert:true})

# upsert false busca un match y si no lo hay no crea un documento con esa
informacion
db.movie.replaceOne({"nombre":"Avengers"}, {"pelicula":"Avengers", "año":
2010}, {upsert:false})

# Metodo Update({}, {})

db.movie.replaceOne({"pelicula":"Avengers"}, {"pelicula":"Avengers", "año":
2019}) # cambiara el documento donde pelicula sea Avengers y le pondra un update
al año

db.movie.replaceOne({"pelicula":"Avengers"}, {$set: {"pelicula":"Avengers",
"año": 2019}}, {upsert:true, multi:true}) # el UPSERT funciona igual pero el
multi es para reemplazar todos los match de documentos que coincidan con pelicula
avengers y actualizarlos ya que por si solo update solo toma el primer match y lo
actualiza y OJO para el multi:true se debe usar {$set:{}} lo que dira que lo que
se introduzca dentro de ese set sera lo que se cambiara.

# Metodo busqueda find

db.movie.find().pretty() #mejor identacion
db.movie.find().count() #cuantas hay
db.movie.find().limit(2) # muestra solo los que quiero
db.movie.find().[2] #estilo array

# Proyecciones

Son un argumento que se les puede aplicar sobre todo a las funciones .find() o
cuando queremos consultar o crear. y sirve para mostrar u ocultar campos, es
similar al SELECT fila1, Fila2...

db.movie.find({"año:2002"}, {"pelicula": 1, "_id": 0, "año": 1}) # nos va a
regresar {"pelicula": "gol", "año": 2002} sin el id

# DeleteOne y many

db.movie.deleteOne({}) # Elimina el primer documento
db.movie.deleteMany({}) # elimina todos los documentos

```

```
db.movie.deleteOne({"_id": ObjectId("1312hjwekjksb")}) # borrar el documnto con  
ese id o puedo filtrarlo por nombres o como yo quiera, pero lo recomendable es  
borrar por ID
```