

CURSO BASE DE DATOS 2 (#13/#14) NoSQL - MongoDB

BIENVENIDAS / BIENVENIDOS

Grupo N3C REM

A/P Jorge Mario Benítez Ruiz,
DSI
Jorge.Benitez@fi365.ort.edu.uy

- Inicio puntual 19:30 hs.
- Es deseable CAMARA ENCENDIDA y
- Se recomienda MICROFONO en Mute al Inicio

MongoDB

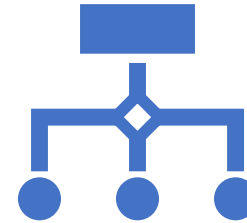
Base de Datos Documental (NoSQL) (Parte C)

MongoDB almacena datos del tipo de formato JSON llamado BSON.

DOCUMENTOS en MongoDB



Los registros de una base de datos MongoDB se denominan documentos,



Los valores de campo pueden incluir números, cadenas, booleanos, matrices o incluso documentos anidados.

Ejemplo:

```
{  
  •title: "Post Title 1",  
  •body: "Body of post.",  
  •category: "News",  
  •likes: 1,  
  •tags: ["news", "events"],  
  •date: Date()  
}
```

SQL vs Document Databases



Las bases de datos SQL se consideran bases de datos relacionales. Almacenan los datos relacionados en tablas separadas. Cuando se necesitan datos, se consultan desde varias tablas para volver a unir los datos.



MongoDB es una base de datos de documentos que a menudo se denomina base de datos no relacional.



MongoDB almacena los datos en documentos flexibles.

Document Databases



MongoDB almacena
datos en documentos
flexibles:



En lugar de tener varias tablas,
simplemente puede mantener
todos los datos relacionados
juntos. Esto hace que la lectura
de sus datos sea muy rápida.







También puede tener varios
grupos de datos. En MongoDB,
en lugar de tablas, se
denominan colecciones.

Arrancando con MONGODB

MongoDB puede instalarse local o hosted en la Nube.

- Descargas sugeridas:

 `mongodb-compass-1.37.0-win32-x64`
 `mongodb-windows-x86_64-6.0.6-signed`
 `mongosh-1.9.1-win32-x64`
 `nosqlbooster4mongo-8.1.1`



Instalaciones de MONGODB

- Puede descargar y utilizar el servidor comunitario de código abierto MongoDB en su hardware de forma gratuita.

[Install MongoDB Community Edition — MongoDB Manual](#)

[NoSQLBooster - The Smartest GUI Tool and IDE for MongoDB](#)

Descargar MongoDB Community

1. Descargue el paquete de instalación Community Server MSI para su sistema operativo:

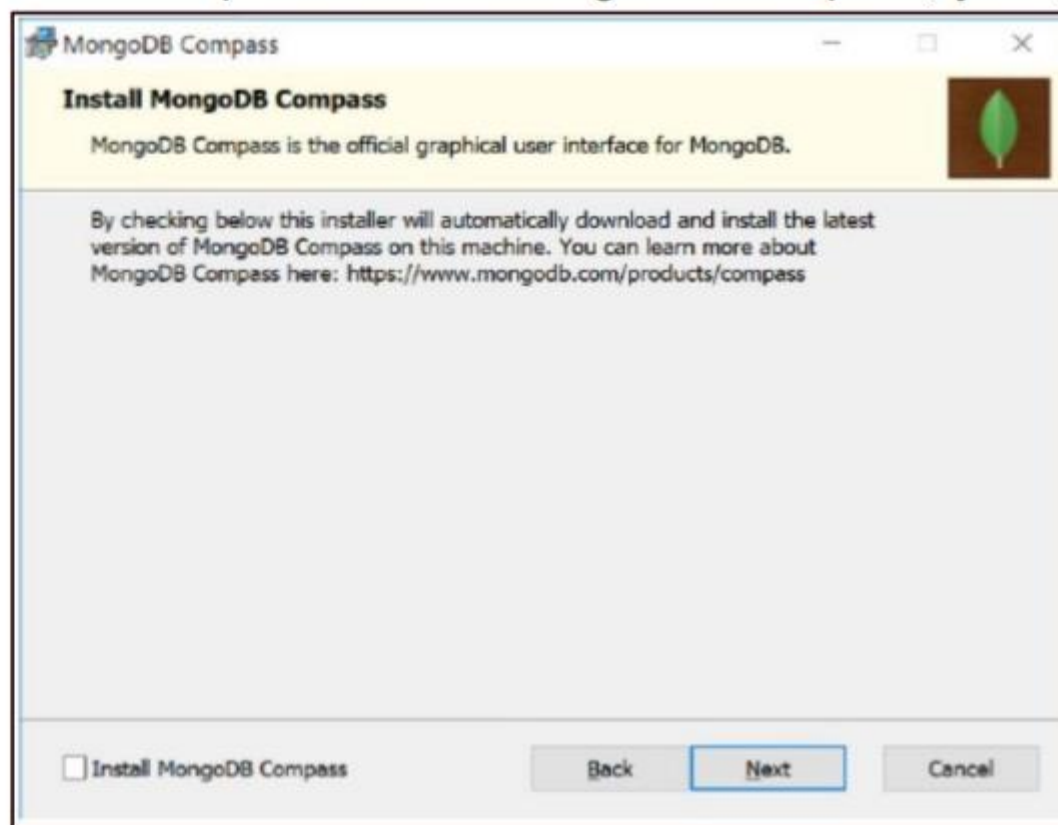
<https://www.mongodb.com/download-center/community>

Version	7.0.2 (current)	▼
Platform	Windows x64	▼
Package	msi	▼
<div>Download </div> <div> Copy link</div> <div>More Options </div>		

Instalar MongoDB Community

1. Ejecute el instalador y haga clic en Next para comenzar.
2. Acepte el acuerdo de licencia y haga clic en Next.
3. Seleccione el tipo de configuración Completa (Complete).
4. Mantenga la opción predeterminada de ejecutar MongoDB como un servicio y haga clic en Next. (Esto permitirá que MongoDB se inicie como un servicio de Windows automáticamente)

5. **DESMARQUE** la casilla para instalar MongoDB Compass, y luego haga clic en Next.



Interfase Grafica para MongoDB:

NoSQLBooster

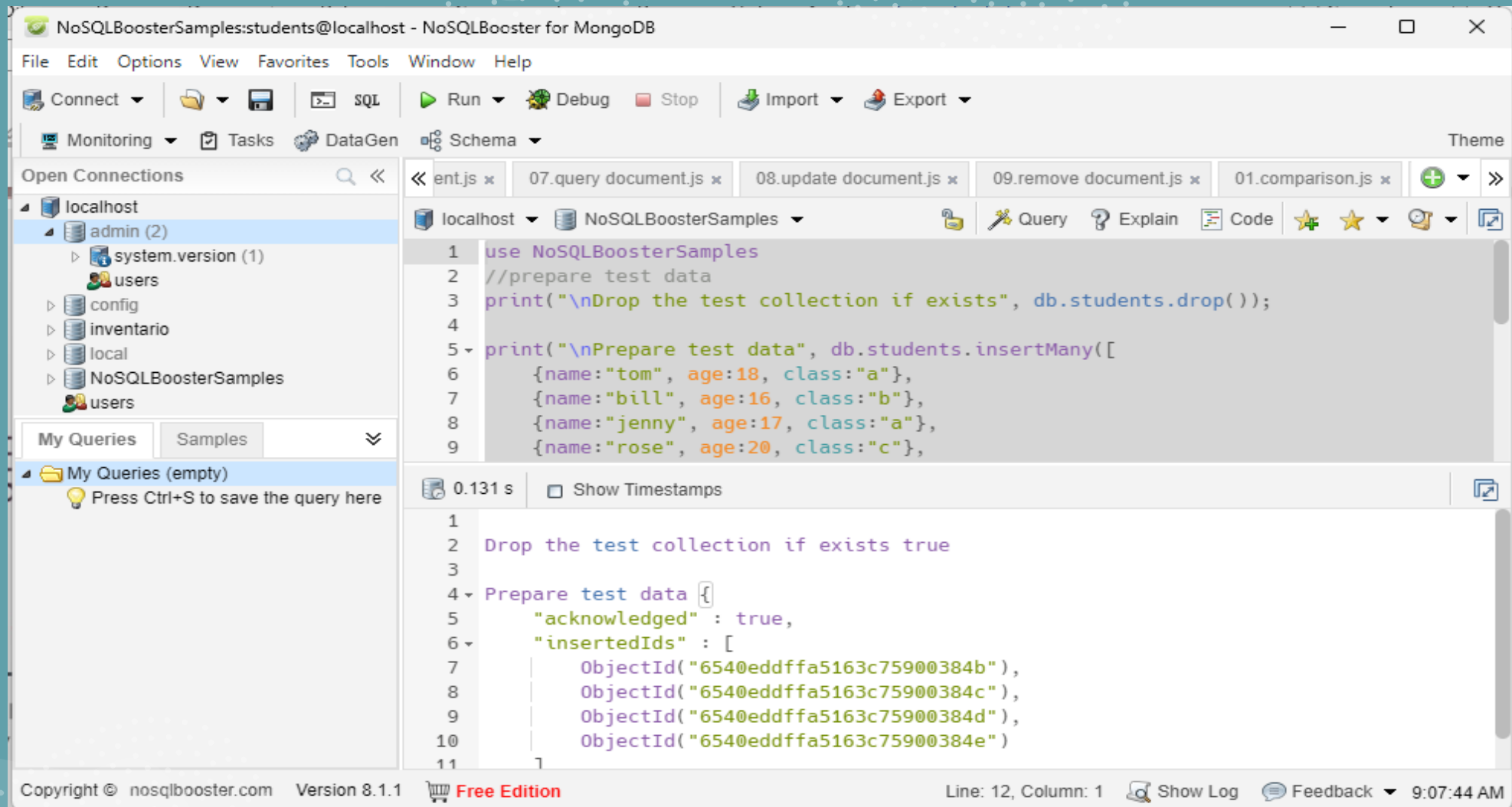
- Descargue e instale el cliente NoSQLBooster aquí:

<https://nosqlbooster.com/>

- Abra la aplicación NoSQLBooster.
- Conéctese a su servidor MongoDB.



NoSQLBooster GUI para MongoDB



MongoDB Create Database

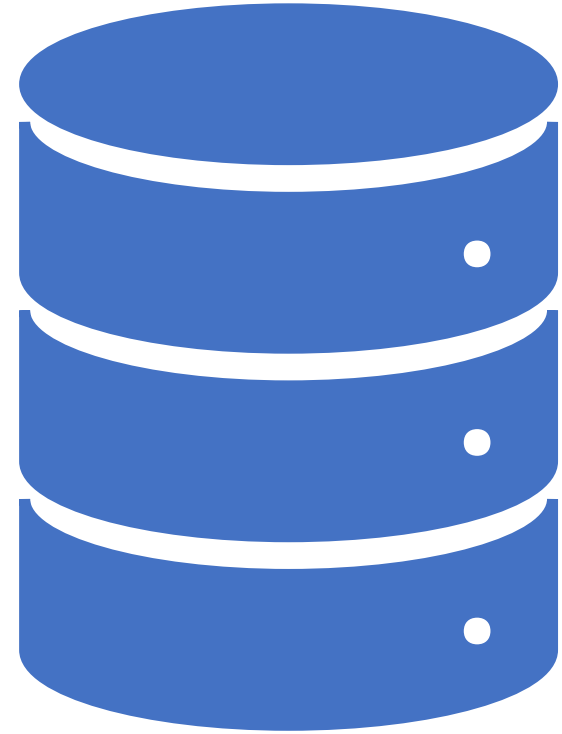
- Create a new database called "**blog**":
- use **blog**
- Ahora estamos en la **blog** database.
- **RECORDAR**: En **MongoDB**, una database NO SE CREA hasta que TIENE CONTENIDO



MongoDB CRUD Operations

CRUD

Create,
Read,
Update,
Delete
documents.



CREATE (Crear Colecciones)

Crear o insertar para agregar nuevos documentos a una colección. Si la colección no existe actualmente, las operaciones de inserción crearán la colección.

```
/* Abrimos la BD (no tiene por que existir) */  
use TestDatabase
```

- `db.collection.insertOne()` *New in version 3.2*
- `db.collection.insertMany()` *New in version 3.2*

In MongoDB, insert operations target a single **collection**. All write operations in MongoDB are **atomic** on the level of a single **document**.

```
db.users.insertOne(  ← collection  
  {  
    name: "sue",      ← field: value  
    age: 26,          ← field: value  
    status: "pending" ← field: value  } document  
  }  
)
```

READ Operations (Operaciones de lectura)

- Las operaciones de lectura recuperan documentos de una colección.
- MongoDB proporciona los siguientes métodos para leer documentos de una colección:

- `db.collection.find()`

You can specify [query filters or criteria](#) that identify the documents to return.

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection
← query criteria
← projection
← cursor modifier

UPDATE Operations (Operaciones de Actualización)

- Las operaciones de actualización modifican los documentos existentes en una colección.
- MongoDB proporciona los siguientes métodos para actualizar los documentos de una colección:

- `db.collection.updateOne()` *New in version 3.2*
- `db.collection.updateMany()` *New in version 3.2*
- `db.collection.replaceOne()` *New in version 3.2*

```
db.users.updateMany(  
  { age: { $lt: 18 } },  
  { $set: { status: "reject" } }  
)
```

← collection
← update filter
← update action

DELETE Operations (Operaciones de Eliminación)

- Eliminar documentos de una colección.
- MongoDB proporciona los siguientes métodos para eliminar documentos de una colección:

- `db.collection.deleteOne()` *New in version 3.2*
- `db.collection.deleteMany()` *New in version 3.2*

```
db.users.deleteMany(  
  { status: "reject" }  
)
```

← collection
← delete filter

SQL to MongoDB Mapping Chart Terminología y conceptos

- SQL to MongoDB Mapping Chart
— MongoDB Manual

Distintos términos y conceptos de SQL y la terminología y los conceptos correspondientes de MongoDB.

SQL Terms/Concepts	MongoDB Terms/Concepts
database	database
table	collection
row	document or BSON document
column	field
index	index
table joins	\$lookup , embedded documents
primary key	primary key
Specify any unique column or column combination as primary key.	In MongoDB, the primary key is automatically set to the _id field.

SQL terminología y MongoDB terminología

SQL INSERT Statements

```
INSERT INTO people(user_id,  
                    age,  
                    status)  
VALUES ("bcd001",  
        45,  
        "A")
```

MongoDB insertOne() Statements

```
db.people.insertOne(  
  { user_id: "bcd001", age: 45, status: "A" }
```

SQL Delete Statements

```
DELETE FROM people  
WHERE status = "D"
```

```
DELETE FROM people
```

MongoDB deleteMany() Statements

```
db.people.deleteMany( { status: "D" } )
```

```
db.people.deleteMany({})
```

SQL SELECT Statements

```
SELECT *  
FROM people
```

```
SELECT user_id, status  
FROM people  
WHERE status = "A"
```

MongoDB find() Statements

```
db.people.find()
```

```
db.people.find(  
  { status: "A" },  
  { user_id: 1, status: 1, _id: 0 }  
)
```

SQL Update Statements

```
UPDATE people  
SET status = "C"  
WHERE age > 25
```

```
UPDATE people  
SET age = age + 3  
WHERE status = "A"
```

MongoDB updateMany() Statements

```
db.people.updateMany(  
  { age: { $gt: 25 } },  
  { $set: { status: "C" } }  
)
```

```
db.people.updateMany(  
  { status: "A" },  
  { $inc: { age: 3 } }  
)
```


OPERADORES de QUERY COMPARACION

Comparison

The following operators can be used in queries to compare values:

- `$eq` : Values are equal
- `$ne` : Values are not equal
- `$gt` : Value is greater than another value
- `$gte` : Value is greater than or equal to another value
- `$lt` : Value is less than another value
- `$lte` : Value is less than or equal to another value
- `$in` : Value is matched within an array

Logical

The following operators can logically compare multiple queries.

- `$and` : Returns documents where both queries match
- `$or` : Returns documents where either query matches
- `$nor` : Returns documents where both queries fail to match
- `$not` : Returns documents where the query does not match

Evaluation

The following operators assist in evaluating documents.

- `$regex` : Allows the use of regular expressions when evaluating field values
- `$text` : Performs a text search
- `$where` : Uses a JavaScript expression to match documents

OPERADORES de UPDATE en MongoDB

Fields

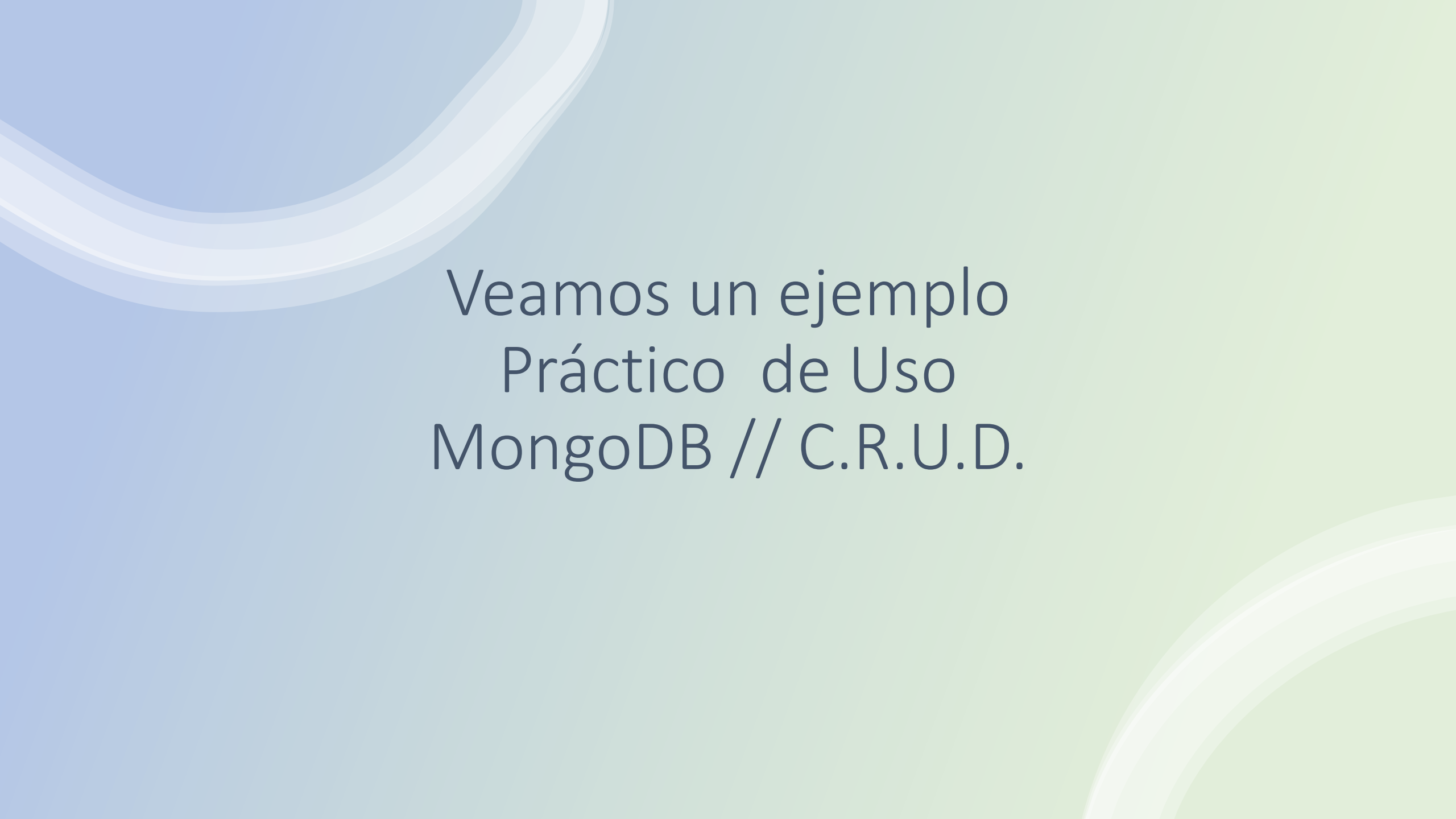
The following operators can be used to update fields:

- `$currentDate` : Sets the field value to the current date
- `$inc` : Increments the field value
- `$rename` : Renames the field
- `$set` : Sets the value of a field
- `$unset` : Removes the field from the document

Array

The following operators assist with updating arrays.

- `$addToSet` : Adds distinct elements to an array
- `$pop` : Removes the first or last element of an array
- `$pull` : Removes all elements from an array that match the query
- `$push` : Adds an element to an array



Veamos un ejemplo
Práctico de Uso
MongoDB // C.R.U.D.



MUCHAS
GRACIAS !!!