

# Proyecto Final NoSQL: Ejercicios MongoDB

---

## Máster Big Data, Data Science & Inteligencia Artificial

Proyecto realizado por: Alejandro Borrego Megías

Fecha: 17 de Noviembre de 2023

Correo: alejbormeg@gmail.com

---

## Table of Contents

- [Introducción a MongoDB](#)
  - [Prerrequisitos](#)
  - [Clientes Utilizados](#)
  - [Ejercicios](#)
- 

## Introducción a MongoDB

MongoDB es un sistema de gestión de bases de datos NoSQL que se ha vuelto ampliamente popular en el ámbito del desarrollo de aplicaciones modernas. A diferencia de las bases de datos relacionales tradicionales, MongoDB utiliza un modelo de datos flexible basado en documentos JSON, lo que permite una escalabilidad horizontal eficiente y una fácil adaptación a los cambios en los esquemas de datos.

## Prerrequisitos

Antes de comenzar con este proyecto, es necesario instalar MongoDB como un servicio de red en su máquina local con sistema operativo Windows. A continuación, se proporcionan los pasos para realizar la instalación:

1. **Descargar MongoDB:** Acceda al [sitio web oficial de MongoDB](#) y descargue la versión Community de MongoDB para Windows.
2. **Instalación:** Siga las instrucciones de instalación proporcionadas durante el proceso de instalación. Asegúrese de seleccionar la opción para instalar MongoDB como un servicio de red.
3. **Configuración:** Después de la instalación, es posible que necesite configurar algunas opciones según sus preferencias. Asegúrese de que el servicio de MongoDB esté iniciado y en ejecución.

## Clientes Utilizados

En este proyecto, se utilizarán dos clientes populares para interactuar con la base de datos MongoDB: Compass y NoSQLBoosterForMongoDB.

1. **MongoDB Compass:** Compass es una interfaz gráfica de usuario que facilita la exploración y manipulación de datos en MongoDB. Proporciona herramientas visuales intuitivas para realizar consultas, analizar el rendimiento y diseñar esquemas de datos.

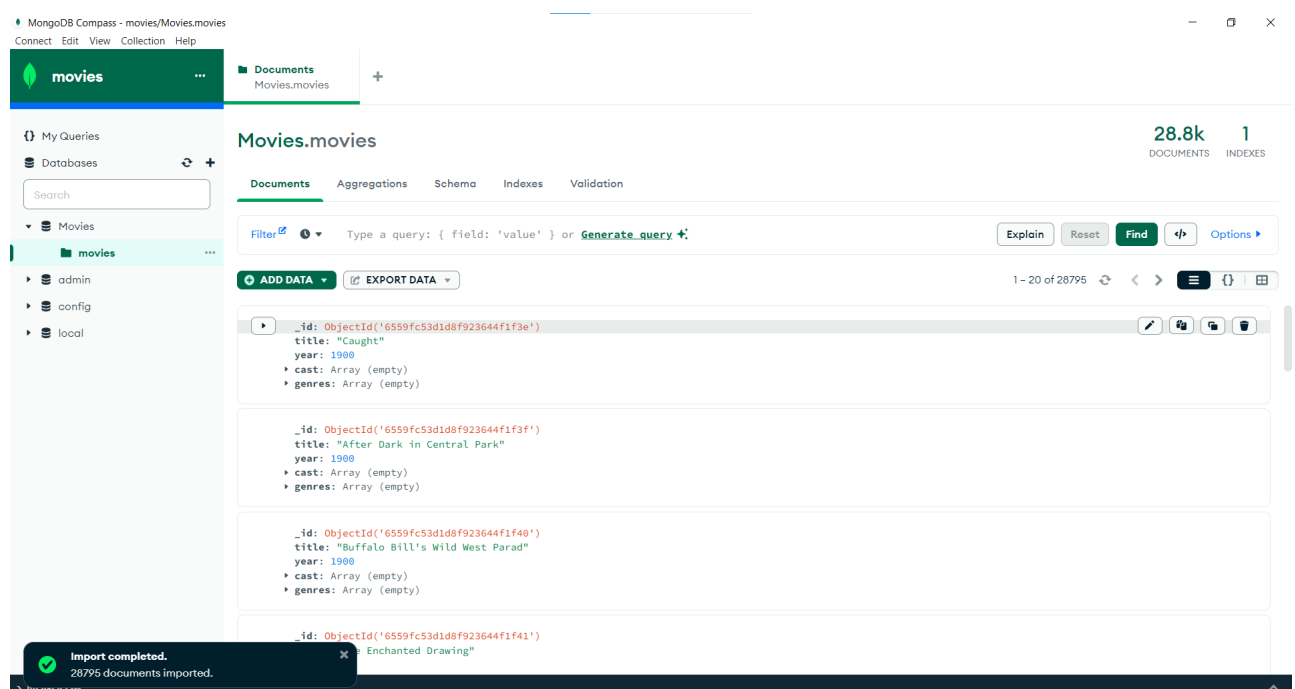
2. **NoSQLBoosterForMongoDB:** NoSQLBooster es otra herramienta poderosa para trabajar con MongoDB. Ofrece características avanzadas como autocompletar consultas, un editor de consultas integrado y una interfaz de usuario amigable que agiliza el desarrollo y la administración de bases de datos MongoDB.

## Ejercicios

A continuación, se presentan los ejercicios que explorarán las capacidades de MongoDB en el contexto de NoSQL. Para el primero usaremos el Cliente *MongoDB Compass* y para el resto de ejercicios el cliente *NoSQLBoosterForMongoDB* por tener una interfaz más amigable para las consultas.

### 0. Realizar la importación del json en una colección llamada "movies"

El Dataset usado para este ejercicio se encuentra en la carpeta `./Database` en formato `.json`. Para importar el *Dataset* usamos el cliente MongoDB Compass:



Como podemos observar se ha importado correctamente.

### 1. Analizar con find la colección.

La query de este ejercicio es la siguiente:

```
// Ejercicio 1
db.movies.find()
```

Con ella podemos ver como en total se han insertado 28795 Documentos en la colección *movies*. Además, vemos como el JSON que representa dichos documentos presenta la siguiente estructura (mostramos el primer ejemplo de la colección):

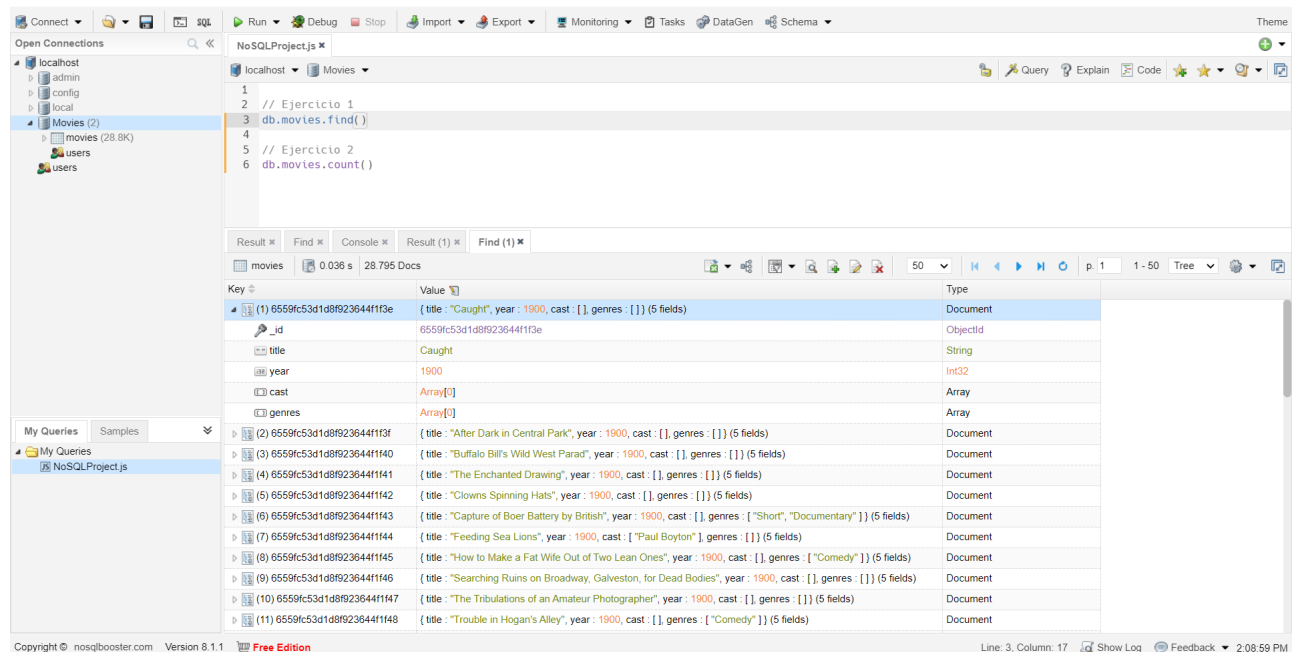
```
{
  "_id" : ObjectId("6559fc53d1d8f923644f1f3e"),
```

```
"title" : "Caught",
"year" : 1900,
"cast" : [ ],
"genres" : [ ]
},
```

Vemos que tiene las siguientes propiedades:

- *title*: es de tipo *string* representando el título de la película.
- *year*: es de tipo *int32* y representa el año de estreno de la película.
- *\*cast*: es de tipo *array* y aunque está vacío se deduce que se incluirán aquí los actores que conforman el casting de la película y quizá información relativa a ellos.
- *generes*: es de tipo *array* y de nuevo en este ejemplo está vacío, pero representará los géneros cinematográficos a los que pertenece dicha película.

A continuación se muestra una captura de pantalla con el resultado obtenido:



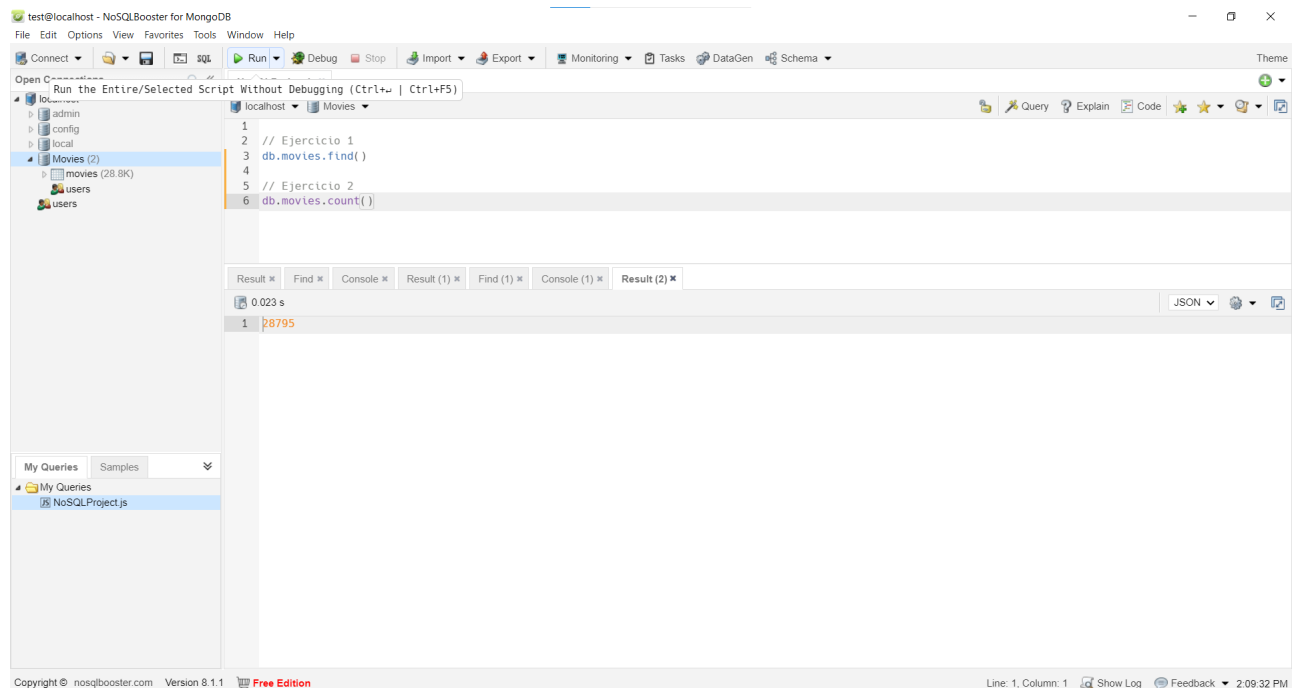
## 2. Contar cuántos documentos (películas) tiene cargado.

La query de este ejercicio es la siguiente:

```
// Ejercicio 2
db.movies.count()
```

El resultado obtenido es que tiene un total de 28795 documentos la colección.

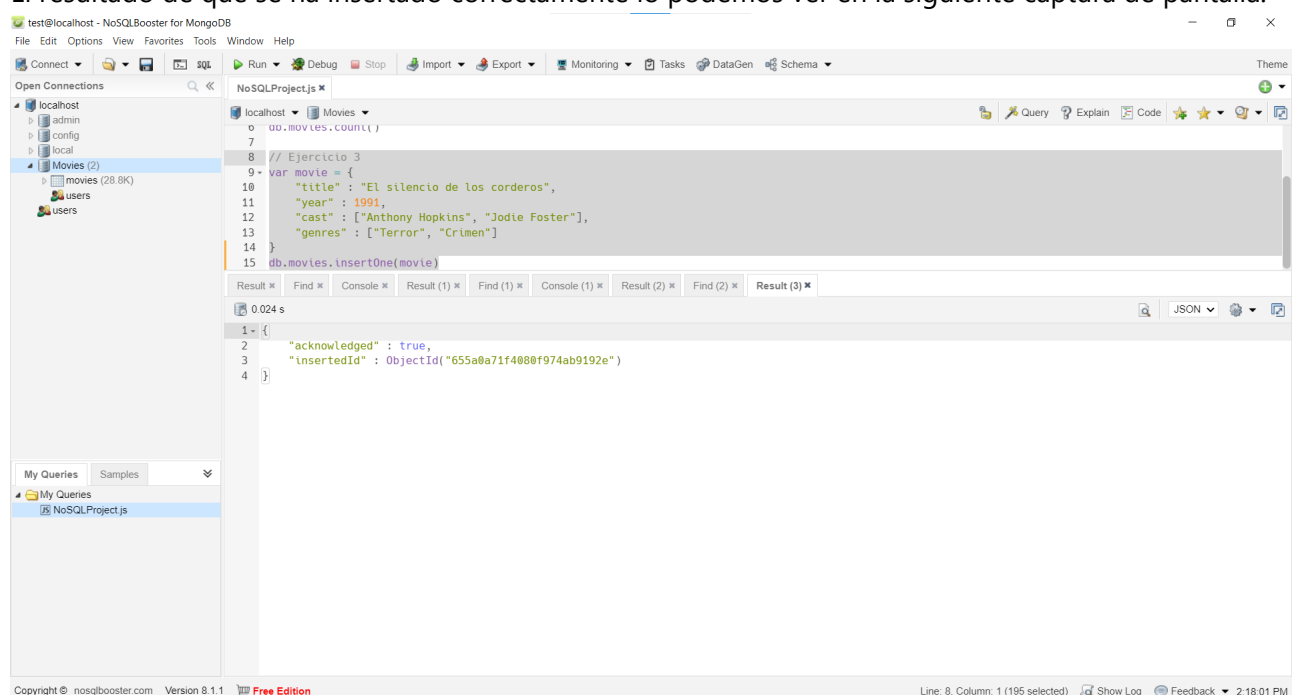
A continuación se muestra una captura de pantalla con el resultado obtenido:



3. **Insertar una película.** La query de este ejercicio es la siguiente:

```
// Ejercicio 3
var movie = {
  "title" : "El silencio de los corderos",
  "year" : 1991,
  "cast" : ["Anthony Hopkins", "Jodie Foster"],
  "genres" : ["Terror", "Crimen"]
}
db.movies.insertOne(movie)
```

El resultado de que se ha insertado correctamente lo podemos ver en la siguiente captura de pantalla:



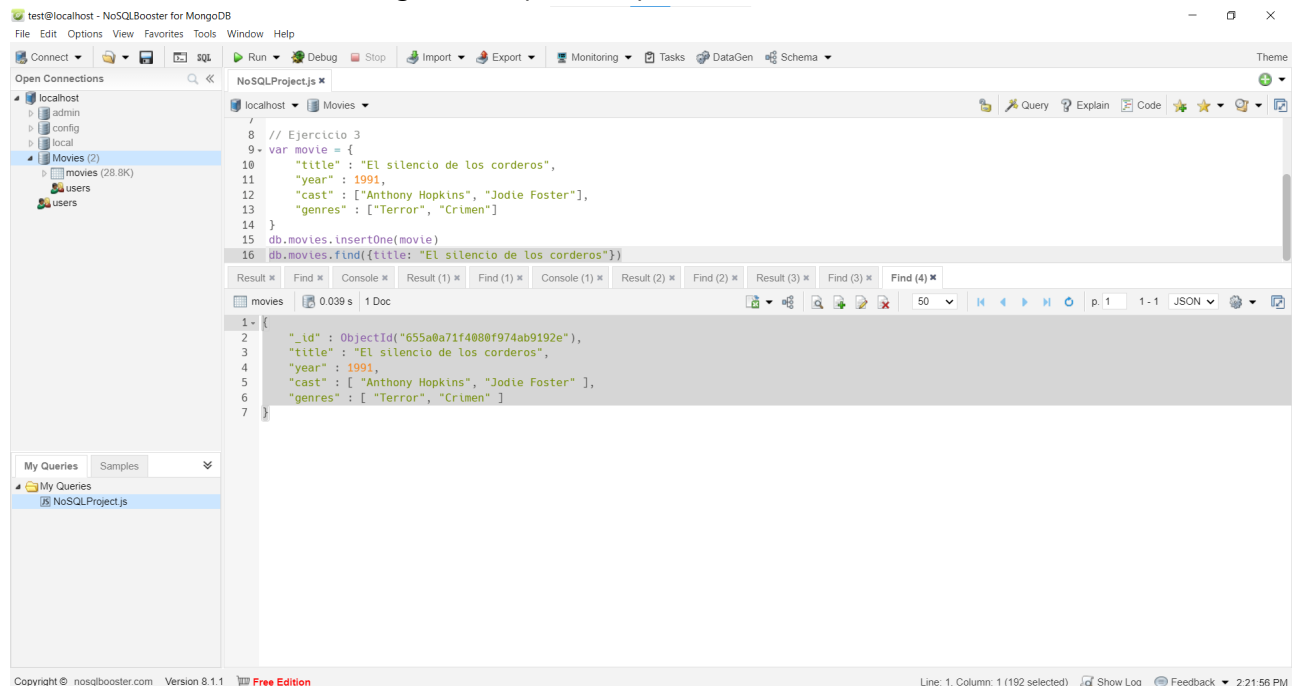
Para comprobar que verdaderamente se ha insertado hacemos una query para buscarla:

```
db.movies.find({title: "El silencio de los corderos"})
```

Obteniendo el siguiente resultado:

```
{
  "_id" : ObjectId("655a0a71f4080f974ab9192e"),
  "title" : "El silencio de los corderos",
  "year" : 1991,
  "cast" : [ "Anthony Hopkins", "Jodie Foster" ],
  "genres" : [ "Terror", "Crimen" ]
}
```

Podemos verlo también en la siguiente captura de pantalla:

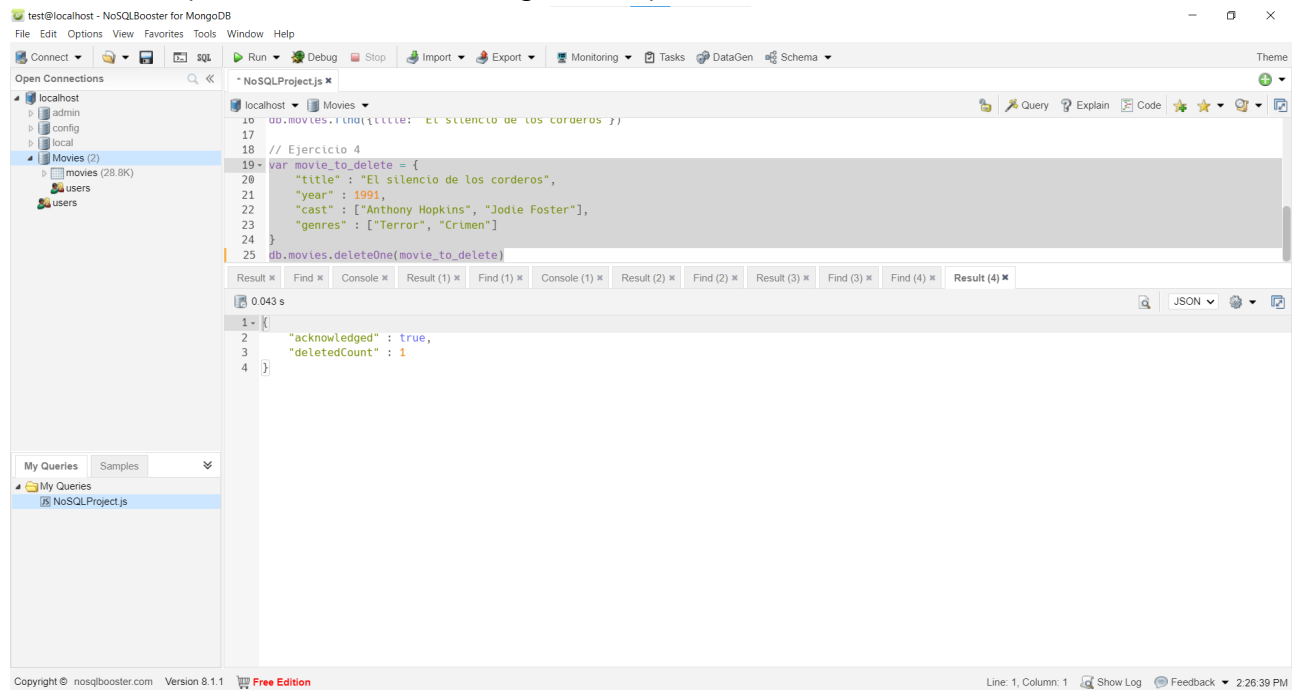


#### 4. Borrar la película insertada en el punto anterior (en el 3).

La query de este ejercicio es la siguiente:

```
var movie_to_delete = {
  "title" : "El silencio de los corderos",
  "year" : 1991,
  "cast" : ["Anthony Hopkins", "Jodie Foster"],
  "genres" : ["Terror", "Crimen"]
}
db.movies.deleteOne(movie_to_delete)
```

Podemos ver que se ha eliminado en la siguiente captura:



Para comprobarlo, ejecutamos la siguiente query:

```
db.movies.find({title: "El silencio de los corderos"})
```

Obteniendo el resultado de un `array` vacío `[]`. Se puede ver en la siguiente captura:

