

# Bases de Datos NoSQL: MongoDB

Autor: José Luis Cendán Guzmán || Correo:[jcendguz@myuax.com](mailto:jcendguz@myuax.com) || Fecha: 20/01/2026

## 1. Informe Microsoft Learn

He completado el módulo **Explore Azure Cosmos DB** de Microsoft Learn, obteniendo una comprensión profunda de las bases de datos NoSQL en la nube y sus aplicaciones prácticas.

### Conceptos Clave Aprendidos

Azure Cosmos DB es un servicio de base de datos NoSQL desarrollado por Microsoft que me ha resultado especialmente interesante por su enfoque global. Lo que más destaca es su capacidad de replicar datos en múltiples regiones del mundo con baja latencia, lo que significa que una aplicación puede servir a usuarios en España, América o Asia con la misma velocidad.

Me ha llamado la atención su flexibilidad para trabajar con diferentes tipos de datos: documentos (como MongoDB), grafos, clave-valor o columnas anchas. Esto permite elegir la estructura más adecuada para cada proyecto sin estar atado a un solo formato. Además, el sistema escala automáticamente según la demanda, ajustando recursos de forma dinámica para optimizar tanto rendimiento como costos.

### Niveles de Consistencia

Uno de los conceptos que más me ha interesado son los cinco niveles de consistencia que ofrece Azure Cosmos DB. Básicamente, permiten elegir entre tener datos siempre actualizados (pero más lento) o aceptar un pequeño retraso a cambio de mayor velocidad.

El nivel **Eventual** es el más rápido porque no espera a que todas las copias se sincronicen, mientras que **Strong** garantiza que siempre lees la versión más reciente pero sacrifica rendimiento. Los niveles intermedios (**Session, Bounded Staleness, Consistent Prefix**) ofrecen diferentes equilibrios según lo que necesite cada aplicación.

### Request Units (RUs)

Las Request Units son la forma en que Azure Cosmos DB mide el costo de las operaciones. En lugar de tener que calcular CPU, memoria y operaciones de disco por separado, todo se unifica en una sola métrica. Esto simplifica mucho la planificación: si una consulta cuesta 10 RUs y otra 100 RUs, sabes que la segunda es 10 veces más costosa, independientemente de qué recursos use internamente.

### Knowledge Check

The screenshot shows the Microsoft Learn module completion page for the 'Explore Azure Cosmos DB' module. At the top, there's a purple banner with the text 'Keep up the great work!' and a circular badge with a blue icon. Below the banner, the module title 'Evaluación de módulos' is displayed, along with a green progress bar indicating '200 XP' earned. On the left, there's a section titled 'Evaluación del módulo superada' with the message 'Buen trabajo. Ha superado la evaluación del módulo.' and a progress bar showing '100%'. To the right, there's an achievement badge for 'Module assessment passed' and the message 'You have earned an achievement! Congratulations, but what should you do next?'. Further down, there's a section titled 'First, let's share your achievement' with social sharing icons for LinkedIn, Email, Twitter, Facebook, and Print. Below that, a call to action 'Don't lose your momentum, keep learning' is shown, followed by a link to 'Next module in this learning path'.

Aquí se puede observar como he completado el módulo y he obtenido una insignia por finalizarlo.

### Reflexión Personal

Lo que más me ha llamado la atención del módulo es la flexibilidad que ofrecen las bases de datos NoSQL frente a las SQL tradicionales. Poder cambiar la estructura de datos sin migraciones complejas es una gran ventaja cuando los requisitos del proyecto evolucionan rápidamente.

También he aprendido que no hay una solución perfecta para todo: hay que elegir entre consistencia y velocidad según cada caso. Las bases de datos NoSQL como Cosmos DB no reemplazan a las relacionales, sino que son complementarias y brillan en escenarios específicos como aplicaciones globales o sistemas con datos muy variables.

## 2. Configuración de MongoDB local

### Bases de datos cargadas

Se han importado exitosamente 7 bases de datos con 24 colecciones. Se pueden observar en MongoDB Express y Compact:

The screenshot shows the MongoDB Express interface. The top navigation bar includes a logo, the text "Mongo Express", and a "Database" dropdown. Below the header, the "Databases" section lists ten databases: admin, config, local, sample\_airbnb, sample\_analytics, sample\_geospatial, sample\_mflix, sample\_supplies, sample\_training, and sample\_weatherdata. Each database entry has a green "View" button on the left and a red "Del" button on the right. A "Create Database" button is located at the top right of this section. The bottom section, titled "Server Status", displays various system metrics in a table format. Key information includes Hostname (ae3c21e09b6), MongoDB Version (8.2.3), Uptime (6945 seconds), Node Version (18.20.3), Server Time (Sun, 18 Jan 2026 23:31:07 GMT), V8 Version (10.2.154.25-node.37), Current Connections (16), Available Connections (419414), Active Clients (0), Queued Operations (0), Clients Reading (0), Clients Writing (0), Read Lock Queue (0), Write Lock Queue (0), Disk Flushes (0), Last Flush (0 ms), Time Spent Flushing (0 ms), Average Flush Time (0 ms), Total Inserts (442411), Total Queries (174), Total Updates (80), and Total Deletes (22).

Databases			
<a href="#">View</a>	admin	<a href="#">Del</a>	
<a href="#">View</a>	config	<a href="#">Del</a>	
<a href="#">View</a>	local	<a href="#">Del</a>	
<a href="#">View</a>	sample_airbnb	<a href="#">Del</a>	
<a href="#">View</a>	sample_analytics	<a href="#">Del</a>	
<a href="#">View</a>	sample_geospatial	<a href="#">Del</a>	
<a href="#">View</a>	sample_mflix	<a href="#">Del</a>	
<a href="#">View</a>	sample_supplies	<a href="#">Del</a>	
<a href="#">View</a>	sample_training	<a href="#">Del</a>	
<a href="#">View</a>	sample_weatherdata	<a href="#">Del</a>	

Server Status			
Hostname	ae3c21e09b6	MongoDB Version	8.2.3
Uptime	6945 seconds	Node Version	18.20.3
Server Time	Sun, 18 Jan 2026 23:31:07 GMT	V8 Version	10.2.154.25-node.37
Current Connections	16	Available Connections	419414
Active Clients	0	Queued Operations	0
Clients Reading	0	Clients Writing	0
Read Lock Queue	0	Write Lock Queue	0
Disk Flushes	0	Last Flush	0 ms
Time Spent Flushing	ms	Average Flush Time	ms
Total Inserts	442411	Total Queries	174
Total Updates	80	Total Deletes	22

Database name	Storage size	Collections	Indexes
admin	40.96 kB	2	3
config	24.58 kB	1	2
local	36.86 kB	3	1
sample_airbnb	53.66 MB	1	1
sample_analytics	9.80 MB	3	3
sample_geospatial	679.94 kB	1	1
sample_mflix	26.80 MB	5	5
sample_supplies	933.89 kB	1	1
sample_training	69.56 MB	9	9
sample_weatherdata	2.63 MB	1	1

Las bases de datos incluyen:

- sample\_airbnb: Listados de AirBnB (5,555 documentos)
- sample\_analytics: Datos financieros (3,992 documentos)
- sample\_geospatial: Datos de naufragios (11,095 documentos)
- sample\_mflix: Películas y cines (75,593 documentos)
- sample\_supplies: Ventas y suministros (5,000 documentos)
- sample\_training: Dataset de entrenamiento variado (331,176 documentos)
- sample\_weatherdata: Datos meteorológicos (10,000 documentos)

### 3. Análisis Agregado 1: Películas por Género

Objetivo: Analizar la distribución de películas por género. Para ello usamos la colección **sample\_mflix.movies**.

Pipeline de Agregación: Dentro de la colección indicada arriba, vamos a la pestaña de agregación y añadimos 3 stages

- **Stage 1: \$unwind.** Lo utilizamos para descomponer la lista de géneros de cada película y poder contarlos individualmente, en lugar de agruparlos por combinaciones.  
\$unwind: { path: "\$genres" }
- **Stage 2: \$group.** Agrupamos las películas por género.  
\$group: {  
  \_id: "\$genres",  
  num\_peliculas: { \$sum: 1 }  
}
- **Stage 3: \$sort.**  
\$sort: { num\_peliculas: -1 }

Proceso en MongoDB Compass

**Stage 1** \$unwind

```
1 ▼ {
2   ▶ path: "$genres"
3 }
```

Output preview after \$unwind stage (Sample of 10 documents)

`_id: ObjectId('573a1390f29313caab50e5')  
plot: "The cartoonist, Winsor McCay, brings  
the Dinosaurus back to life in the..."  
genres: "Animation"  
runtime: 12  
cast: Array (3)  
num_mflix_comments: 1  
poster: "https://m.media-  
amazon.com/images/M/MV5BMTQxNzI40D...`

`_id: ObjectId('573a1390f29313caab50e5')  
plot: "The cartoonist, Winsor McCay, brings  
the Dinosaurus back to life in the..."  
genres: "Short"  
runtime: 12  
cast: Array (3)  
num_mflix_comments: 1  
poster: "https://m.media-  
amazon.com/images/M/MV5BMTQxNzI40D...`

`_id: ObjectId('573a1390f29313caab50e5')  
plot: "The cartoonist, Winsor McCay, brings  
the Dinosaurus back to life in the..."  
genres: "Comedy"  
runtime: 12  
cast: Array (3)  
num_mflix_comments: 1  
poster: "https://m.media-  
amazon.com/images/M/MV5BMTQxNzI40D...`

---

**Stage 2** \$group

```
1 ▼ {
2   ▶ _id: "$genres",
3   // ahora genres es un STRING
4   ▶ num_peliculas: {
5     $sum: 1
6   }
7 }
```

Output preview after \$group stage (Sample of 10 documents)

`_id: "Action"  
num_peliculas: 2539`

`_id: "Crime"  
num_peliculas: 2678`

`_id: "News"  
num_peliculas: 51`

---

**Stage 3** \$sort

```
1 ▼ {
2   num_peliculas: -1
3 }
```

Output preview after \$sort stage (Sample of 10 documents)

`_id: "Drama"  
num_peliculas: 13789`

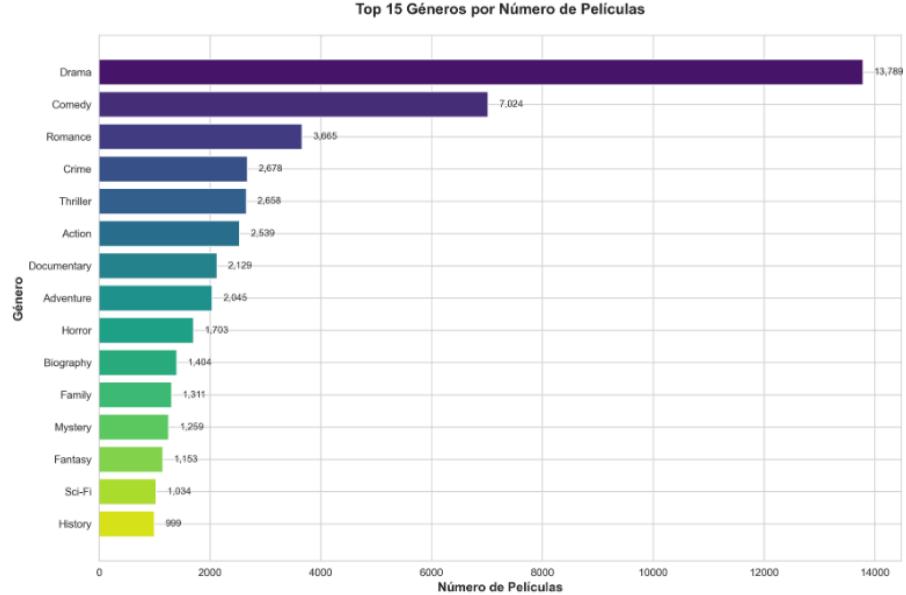
`_id: "Comedy"  
num_peliculas: 7024`

`_id: "Romance"  
num_peliculas: 3665`

## Resultados

Los 5 géneros principales son:

Género	Nº de películas
Drama	13.789
Comedy	7.024
Romance	3,665
Crime	2,678
Thriller	2,658



## Análisis

El género **Drama** domina claramente con casi el doble de películas que Comedy. Los géneros narrativos (Drama, Comedy, Romance) son los más prolíficos, mientras que géneros de nicho tienen menor representación.

#### 4. Análisis Agregado 2: Inspecciones por Resultado

Objetivo: Analizar la distribución de inspecciones de negocios por resultado en la colección `sample_training.inspections`.

##### Pipeline de Agregación

- Stage 1 : \$group: Agrupar inspecciones por resultado
  - \$group: {
    - \_id: "\$result",
    - num\_inspecciones: { \$sum: 1 }
- Stage2: \$sort
  - \$sort: { num\_inspecciones: -1 }

##### Proceso en MongoDB Compass

The screenshot shows the MongoDB Compass interface with two stages of an aggregation pipeline. Stage 1 (\$group) groups documents by result type and counts them. Stage 2 (\$sort) sorts the grouped documents by the count in descending order. The output preview shows three results from each stage.

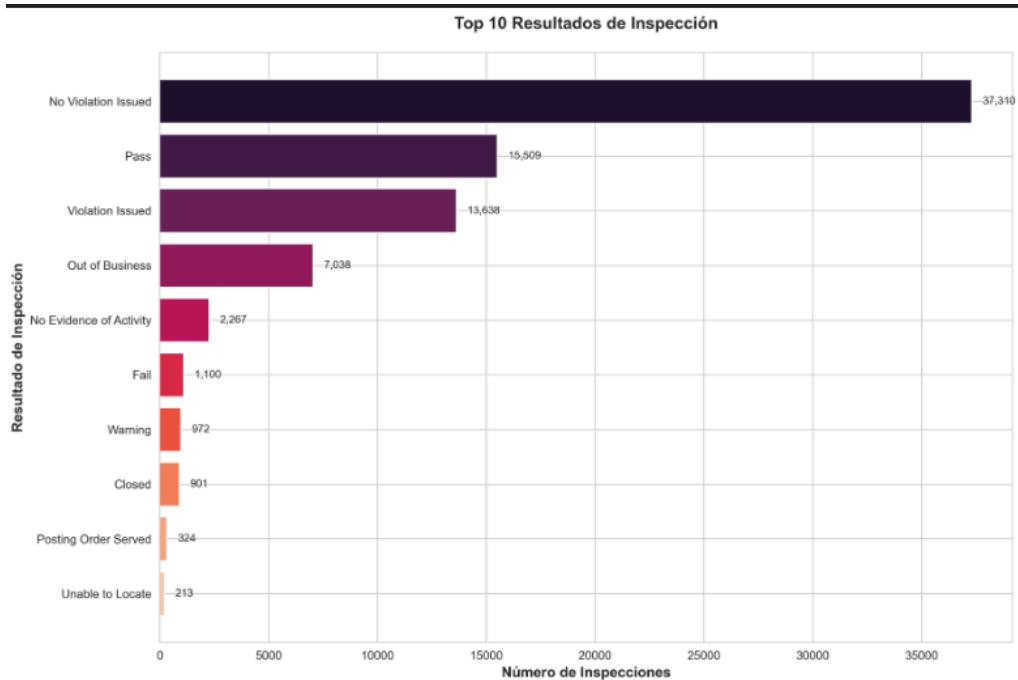
Resultados Stage 1 (\$group)	Output preview after \$group stage (Sample of 10 documents)
<pre>1 ▼ { 2   _id: "\$result", 3   num_inspecciones: { 4     \$sum: 1 5   } 6 }</pre>	<pre>_id: "Warning" num_inspecciones : 972</pre> <pre>_id: "No Evidence of Activity" num_inspecciones : 2267</pre> <pre>_id: "Violation Issued" num_inspecciones : 13638</pre>

Resultados Stage 2 (\$sort)	Output preview after \$sort stage (Sample of 10 documents)
<pre>1 ▼ { 2   num_inspecciones: -1 3 }</pre>	<pre>_id: "No Violation Issued" num_inspecciones : 37310</pre> <pre>_id: "Pass" num_inspecciones : 15509</pre> <pre>_id: "Violation Issued" num_inspecciones : 13638</pre>

##### Resultados

Los 5 resultados principales son:

Resultado	Nº Inspecciones
No Violation Issued	37,310
Pass	15,509
Violation Issued	13,638
Out of Business	7,038
No Evidence of Activity	2,267



## Análisis

La mayoría de inspecciones (47%) resultan en **No Violation Issued**, indicando que la mayoría de negocios cumplen con las regulaciones. Sin embargo, un 17% recibe violaciones, lo que sugiere la necesidad de seguimiento continuo.

Ambas agregaciones, se encuentran ahora en my queries:

The screenshot shows two aggregate queries in MongoDB Compass:

- .AGGREGATE** **Inspecciones por resultado**
  - sample\_training
  - inspections

Last modified: 3 minutes ago
- .AGGREGATE** **Películas por género**
  - sample\_mflix
  - movies

Last modified: 2 minutes ago

## Código de Generación

Los gráficos fueron generados con Python usando matplotlib y seaborn:

```
import json
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar datos
with open('sample_mflix.movies.json', 'r') as f:
    data = json.load(f)

# Crear gráfico de barras horizontales
plt.barh(generos, num_peliculas)
plt.xlabel('Número de Películas')
plt.ylabel('Género')
plt.title('Top 15 Géneros por Número de Películas')
plt.savefig('peliculas_por_genero.png', dpi=300)
````
```

## 5. Conclusiones

Esta práctica ha supuesto una evolución importante en mi forma de concebir el almacenamiento y análisis de datos. Acostumbrado a la estructura rígida de las bases de datos relacionales, trabajar con MongoDB me ha mostrado el valor real de la flexibilidad de esquema. La capacidad de ingerir y analizar datos semi-estructurados sin necesidad de definir

previamente una estructura rígida agiliza enormemente las fases iniciales de cualquier proyecto de análisis, permitiendo dedicar más tiempo a la extracción de valor y menos a la gestión de la infraestructura.

Desde una perspectiva analítica, el **Framework de Agregación** me ha parecido una herramienta excepcionalmente potente. Operadores como **\$unwind** permiten realizar transformaciones complejas sobre los datos de forma nativa y eficiente, resolviendo en pocos pasos problemas que en SQL implicaría múltiples uniones costosas. Además, la posibilidad de visualizar el flujo de datos paso a paso en MongoDB Compass facilita la comprensión profunda de cómo se transforman y agregan las métricas, algo esencial para garantizar la calidad del análisis.

En conclusión, considero que NoSQL y SQL son paradigmas complementarios. Mientras que el modelo relacional sigue siendo insustituible para garantizar la integridad en transacciones complejas, tecnologías como Azure Cosmos DB y MongoDB se perfilan como soluciones ideales para escenarios de Big Data, análisis en tiempo real y aplicaciones que requieren escalabilidad global. Incorporar estas herramientas a mi stack tecnológico me permite abordar problemas de datos más diversos con la solución más adecuada para cada caso.