



mongoDB®

MONGODB ATLAS

Integrantes:

Ambrosio Reyes Jorge Alejandro

Castillo Antonio José

Cruz Esquivel Christian

¿QUÉ ES MONGODB?

- Es un Sistema Gestor de Base de Datos (SGBD) no relacional (noSQL)
- Open-Source
- Orientado a Documentos
- Multiplataforma
- High Performance -> Alto Rendimiento
- High Availability -> Alta Disponibilidad
- Alta Escalabilidad

¿QUÉ ES MONGODB?

- MongoDB almacena datos en documentos JSON flexibles, es decir, cada documento puede contener diferentes campos y las estructuras de datos se pueden ir modificando.
- El modelo de documentos concuerda con los objetos del código de la aplicación, lo que facilita trabajar con datos.
- Las consultas ad-hoc, la indexación y la agregación en tiempo real permiten acceder a los datos y analizarlos con gran eficacia.
- MongoDB es una base de datos distribuida, por lo que es fácil de usar y proporciona una elevada disponibilidad, escalabilidad horizontal y distribución geográfica.

¿CÓMO FUNCIONA MONGODB?

- El modelo de documentos de MongoDB resulta muy fácil de aprender y usar, y proporciona a los desarrolladores todas las funcionalidades que necesitan para satisfacer los requisitos más complejos a cualquier escala. Se proporcionan drivers para más de diez lenguajes, y la comunidad ha desarrollado varias decenas más.

¿CÓMO FUNCIONA MONGODB?

EJEMPLO EN JAVA (CONEXIÓN):

- Aquí se realiza la conexión con una base de datos de MongoDB alojada localmente denominada test con una colección llamada restaurants:

```
// 1. Conexión a la instancia MongoDB ejecutándose en localhost
MongoClient = new MongoClient();
```

```
// Acceso a base de datos llamada 'test'
MongoDatabase database =
mongoClient.getDatabase("test");
```

```
// Acceso a la colección llamada 'restaurantes'
MongoCollection collection =
database.getCollection("restaurants");
```

¿CÓMO FUNCIONA MONGODB? EJEMPLO EN JAVA (INSERTAR DOCUMENTO):

- Aquí e están insertando 2 documentos de ejemplo en la restaurants colección. Cada documento representa un restaurante con un nombre, clasificación por estrellas y categorías (almacenados como una matriz):

```
// 2. Insertar
List documents = asList(
    new Document("name", "Sun Bakery Trattoria")
        .append("stars", 4)
        .append("categories",
            asList("Pizza", "Pasta", "Italian", "Coffee", "Sandwiches")),
    new Document("name", "Blue Bagels Grill")
        .append("stars", 3)
        .append("categories",
            asList("Bagels", "Cookies", "Sandwiches")));
collection.insertMany(documents);
```

¿CÓMO FUNCIONA MONGODB? EJEMPLO EN JAVA (CREAR CONSULTA):

- En este ejemplo, ejecutamos una consulta sencilla para obtener todos los documentos en la restaurants colección y los guardamos como una matriz:

```
// 3. Consulta  
List results = collection.find().into(new ArrayList<>());
```


¿CÓMO FUNCIONA MONGODB? EJEMPLO EN JAVA (CREAR ÍNDICE):

- Los índices de MongoDB son similares a los de otros sistemas de bases de datos. MongoDB soporta índices en cualquier campo o subcampo de un documento perteneciente a una colección. Aquí, estamos construyendo un índice en el campo de nombre con orden de clasificación ascendente.

```
// 4. Crear índice  
collection.createIndex(Indexes.ascending("name"));
```


¿CÓMO FUNCIONA MONGODB? EJEMPLO EN JAVA (REALIZAR AGREGACIÓN):

- Con el pipeline de agregación de MongoDB, puede filtrar y analizar datos en base a un conjunto determinado de criterios. En este ejemplo, extraemos todos los documentos de la colección restaurantes con categoría de panadería mediante el \$match operador y luego los agrupamos según el número de estrellas de su puntuación, utilizando el \$group operador. Con el operador acumulador, \$sum podemos ver cuántas panaderías de nuestra colección tienen el mismo número de estrellas.

```
// 5. Realizar agregación
    collection.aggregate(asList(match(eq("categories", "Bakery")),
    group("$stars", sum("count", 1))));

    mongoClient.close();
```

¿COMO FUNCIONA?

- MongoDB tiene la siguiente estructura:

MongoServer -> Base de Datos -> Colecciones -> Documentos -> Pares de cadena y valor (JSON)

```
Ej {  
  "atributo1": "valor1",  
  "atributo2": "valor2",  
  "atributo3": "true"  
}
```

COSTOS DE MONGODB

	Free	Basic	RECOMMENDED Pro	Enterprise
	For learning, prototypes, and early development	Designed for every stage of your application	Includes proactive support for mission-critical workloads	Includes proactive support and advanced enterprise features
	Shared RAM	Scalable RAM	Scalable RAM	Scalable RAM
	512 MB storage	Scalable Storage	Scalable Storage	Scalable Storage
	Get started free	From \$9/mo	Contact us	Contact us
Cluster Customization				
Highly available clusters	●	●	●	●
Uptime SLA		●	●	●
Elastic scalability		●	●	●
Cross-region replication		●	●	●
Global clusters		●	●	●

COSTOS DE MONGODB

Security				
Always-on authentication	●	●	●	●
IP whitelists	●	●	●	●
End-to-end encryption	●	●	●	●
Dedicated private network		●	●	●
Enterprise security features				●
Performance Optimization				
Monitoring and alerts	●	●	●	●
100+ metrics dashboard		●	●	●
Real-time performance panel		●	●	●
Performance advisor		●	●	●
Disaster Recovery				
Fully managed backups		●	●	●
Point-in-time recovery		●	●	●
Queryable snapshots		●	●	●

COSTOS DE MONGODB

Support & Learning			
Platform and UI support	●	●	●
End-to-end database support	30 days free	30 days free	●
Response time SLA		2 hrs	1 hr
On-demand access to MongoDB University			●
Additional Software & Tools			
MongoDB Compass		●	●
MongoDB Connector for BI			●

Need a faster response time SLA? [Contact us.](#)

NOSQL: ¿CUANDO ES RECOMENDABLE USARLAS?

- Si la aplicación va a tener un crecimiento muy rápido
- Si va a tener servidores en la nube
- Al querer montar una base de datos lo mas rápido posible
- Si los datos no van a tener siempre la misma estructura
- La información será muy dinámica
- Cuando se tendrá demasiados usuarios accediendo al mismo tiempo

DOCUMENTOS

- Son una abstracción/descripción de una entidad/objeto de la vida Real
- Como se menciona, se basa en el formato JSON (JavaScript Object Notation)

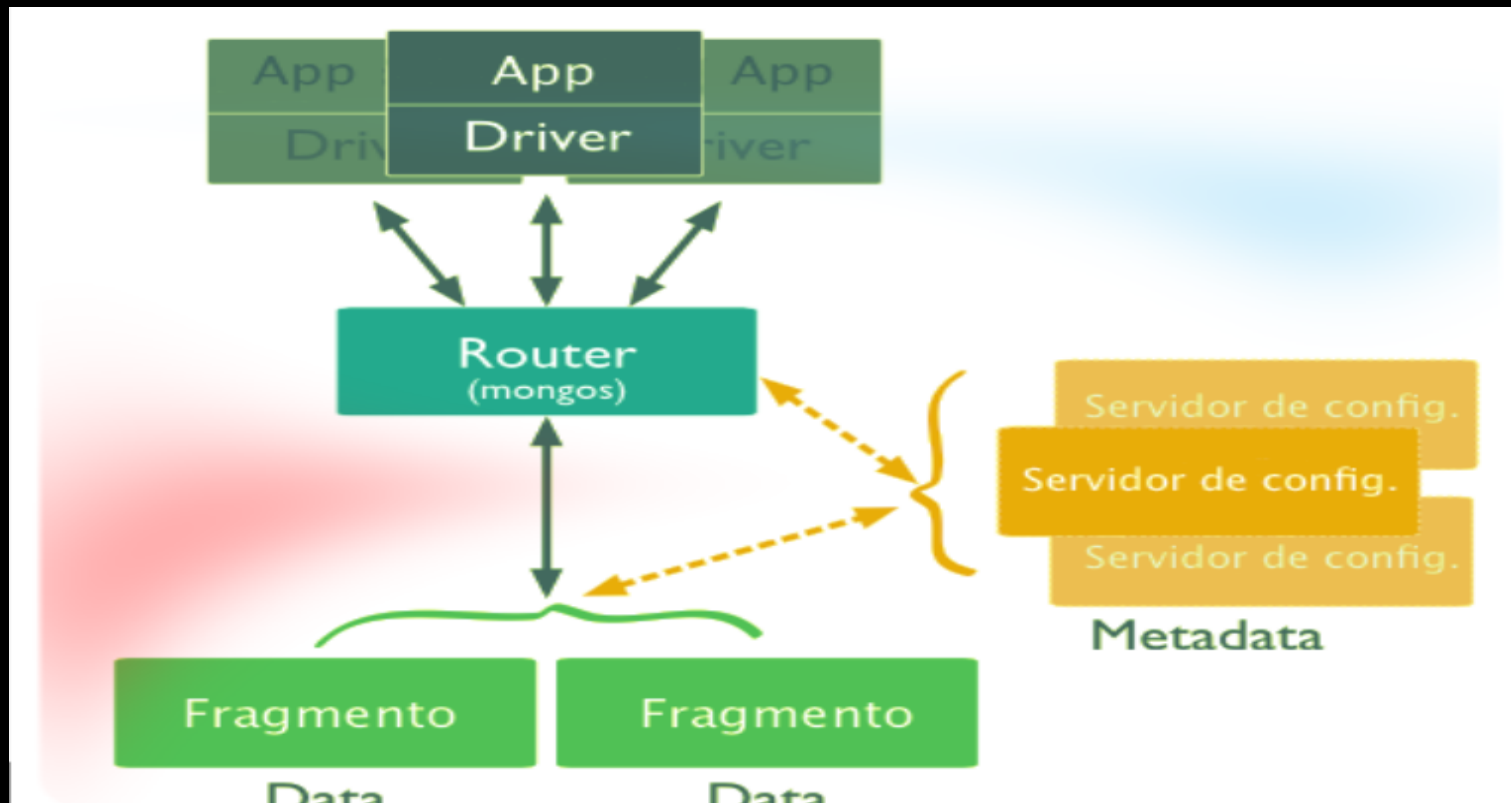
- Definición de un documento para una persona:

```
{“Nombre”:”Cosme”,  
  “Apellido”:”Fulanito”,  
  “id”:”090265”,  
  “edad”:28;  
}
```


COLECCIONES

- Representan un conjunto de documentos de una misma entidad
- Una colección va a tener TODOS los usuarios, Personas, Productos, Aeropuertos, etc.

ARQUITECTURA DE MONGODB



APLICACIÓN Y DRIVER

- Las Aplicaciones, cuando necesitan comunicarse con la base de datos de MongoDB, lo hacen a través de un driver, estos tienen implementados los métodos y protocolos necesarios para comunicarse correctamente con la base de datos encapsulando la complejidad del proceso al desarrollador

FRAGMENTOS

- Son los que poseen los datos fragmentados de las colecciones que componen a la base de datos como tal
- Este suele estar compuesto por una replica , sin embargo, en ambientes de desarrollo podría ser una única instancia

ROUTER

- Debido a que las aplicaciones ven la base de datos como un todo, el router es el encargado de recibir las peticiones y dirigir las operaciones necesarias al fragmento o fragmentos correspondientes

SERVIDOR DE CONFIGURACIÓN (METADATA)

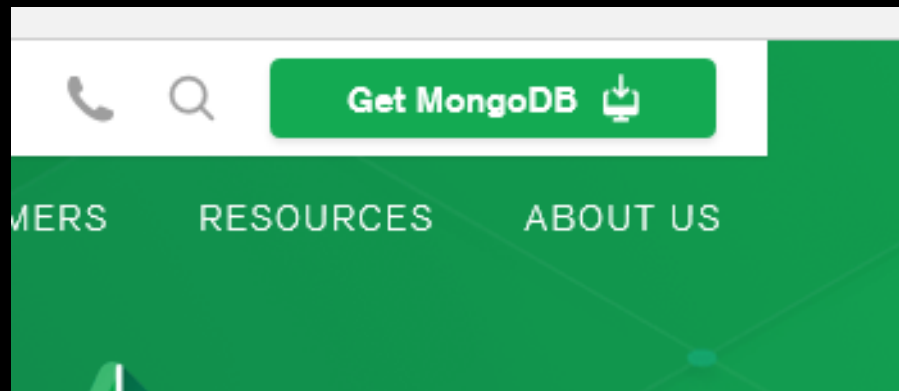
- Este tipo de instancia se encarga de almacenar la metadata de la fragmentación.
- Esta información es almacenada en caché por el Router para lograr un óptimo tiempo de procesamiento

MONGODB ATLAS

- MongoDB Atlas ofrece la base de datos líder en el mundo para aplicaciones modernas como un servicio en la nube totalmente automatizado diseñado y administrado por el mismo equipo que construye la base de datos.
- MongoDB Atlas automatiza el aprovisionamiento, la configuración y el despliegue de la infraestructura para que los equipos puedan obtener los recursos de la base de datos que necesitan, cuando son necesarios.
- Los clústeres de recuperación automática están formados por instancias de bases de datos distribuidas geográficamente para garantizar que no haya un punto único de falla. Incluso puede replicar la base de datos en múltiples regiones de la nube para una tolerancia de fallas de varias regiones.
- MongoDB Atlas está disponible con soporte para más de 50 regiones en la nube en Amazon Web Services, Microsoft Azure y Google Cloud Platform.

CREACIÓN DE CUENTA EN MONGODB

- Ir a la dirección <https://www.mongodb.com/>
- Hacer click en Obtener MongoDB, en la esquina superior derecha:



CREACIÓN DE CUENTA EN MONGODB

- Llenar el formulario que aparecerá a continuación:

No download necessary
Deploy a free cluster now

Email address

First name Last name

Password

- ✓ 8 character minimum
- ✓ One number
- ✓ One letter
- ✓ One special character

I agree to the [terms of services](#).

Get started free

CREACIÓN DE CUENTA EN MONGODB

- Listo ya se puede comenzar a configurar y utilizar:

mongoDB Atlas




CLUSTERS > CREATE NEW CLUSTER

Create New Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).

Global Cluster Configuration >

Cloud Provider & Region AWS, N. Virginia (us-east-1) ▾

Create a **free tier cluster** by selecting a region with **FREE TIER AVAILABLE** and choosing the **M0** cluster tier below.

★ recommended region ⓘ

CONFIGURACIÓN DE CLUSTERS

Cluster Tier

M30 (8 GB RAM, 40 GB Storage) 

120 IOPS, Encrypted, Auto-expand Storage

Base hourly rate is for a MongoDB replica set with 3 data bearing servers.


Shared Clusters

M0	Shared RAM	512 MB Storage	Shared vCPUs	Free forever
M2	Shared RAM	2 GB Storage	Shared vCPUs	\$9 / MONTH
M5	Shared RAM	5 GB Storage	Shared vCPUs	\$25 / MONTH

Dedicated Development Clusters

M10	2 GB RAM	10 GB Storage	0.2 vCPUs	from \$0.08/hr
M20	4 GB RAM	20 GB Storage	0.4 vCPUs	from \$0.20/hr

Dedicated Production Clusters

 M30	8 GB RAM	40 GB Storage	2 vCPUs	from \$0.54/hr
> M40	16 GB RAM	80 GB Storage	4 vCPUs	from \$1.04/hr
> M50	32 GB RAM	160 GB Storage	8 vCPUs	from \$2.00/hr

CONFIGURACIÓN DE CLUSTERS

SANDBOX

Cluster0
Version 4.0.4

[CONNECT](#) [METRICS](#) [COLLECTIONS](#) [...](#)

INSTANCE SIZE
M0

REGION
AWS / N. Virginia (us-east-1)

TYPE
Replica Set - 3 nodes

LINKED STITCH APP
None Linked - [Link Application](#)

Operations R: 0 W: 0 100.0/s

Last 6 Hours

Logical Size 0.0 B 512.0 MB max

Last 6 Hours

Connections 0 100 max

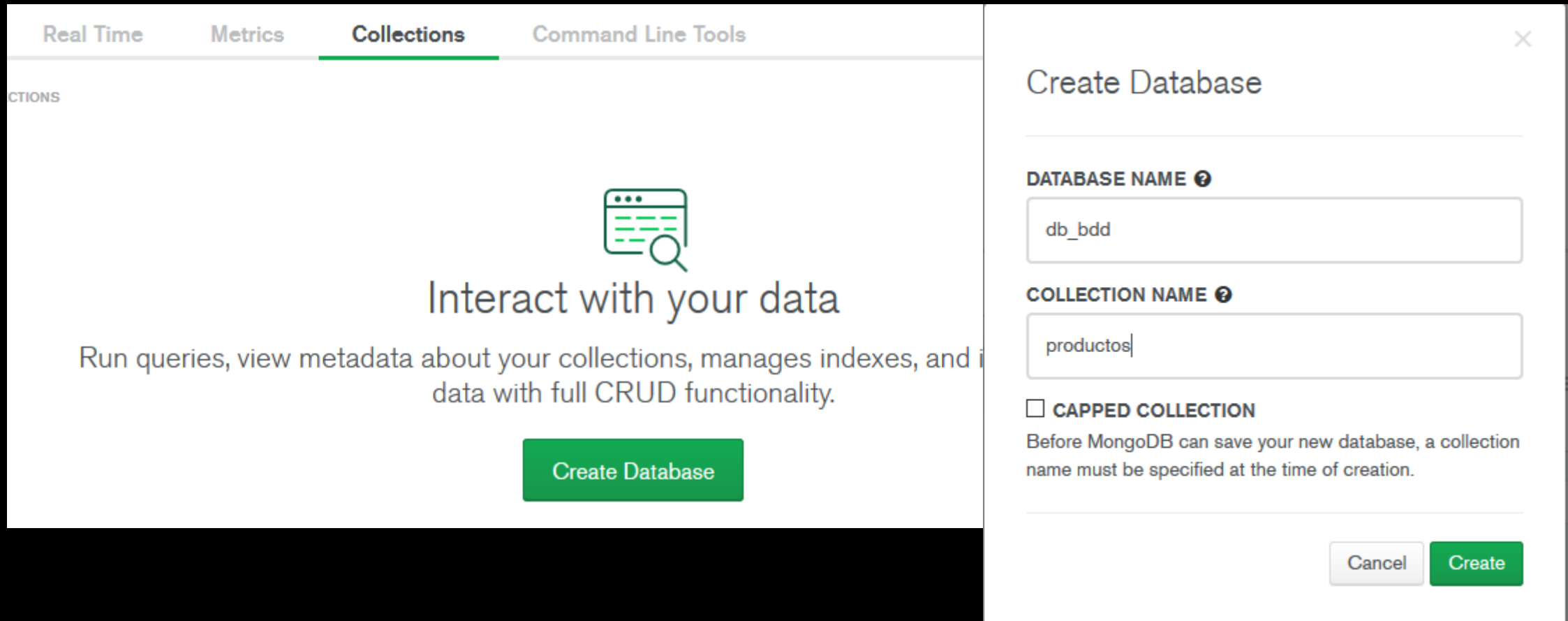
Last 6 Hours

Enhance Your Experience

For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

[Upgrade](#)


CREACIÓN DE BASES DE DATOS



The image shows a screenshot of the MongoDB Compass web interface. The top navigation bar includes 'Real Time', 'Metrics', 'Collections', and 'Command Line Tools'. The 'Collections' tab is active. The main content area displays a large green button labeled 'Create Database'. A modal dialog box titled 'Create Database' is open on the right side. The dialog contains two input fields: 'DATABASE NAME' with the value 'db_bdd' and 'COLLECTION NAME' with the value 'productos'. There is an unchecked checkbox for 'CAPPED COLLECTION' with a descriptive note below it. At the bottom of the dialog are 'Cancel' and 'Create' buttons.

Real Time Metrics **Collections** Command Line Tools

CTIONS



Interact with your data

Run queries, view metadata about your collections, manages indexes, and i
data with full CRUD functionality.

Create Database

Create Database

DATABASE NAME ⓘ

db_bdd

COLLECTION NAME ⓘ

productos|

CAPPED COLLECTION
Before MongoDB can save your new database, a collection name must be specified at the time of creation.

Cancel **Create**

CREACIÓN DE ELEMENTOS

The screenshot displays the MongoDB Compass interface. At the top, there are navigation tabs: Overview, Real Time, Metrics, Collections (which is active and highlighted with a green underline), and Command Line Tools. Below the tabs, the interface shows '2 DATABASES 3 COLLECTIONS' and a '+ Create Database' button. A search bar for 'NAMESPACES' is present. On the left sidebar, a tree view shows the database structure: 'db_bdd' (with a sub-item 'productos') and 'db_test' (with sub-items 'alumnos' and 'ueas', where 'ueas' is selected). The main content area is titled 'db_test.ueas' and displays 'COLLECTION SIZE 0B', 'TOTAL DOCUMENTS 0', and 'INDEXES TOTAL SIZE 4KB'. There is a '+ INSERT DOCUMENT' button. Below this, there are 'Find' and 'Indexes' tabs, with 'Find' selected. A search bar contains a filter query: `{"filter": "example"}`, with 'FILTER' and 'Find' buttons. Below the search bar, it shows 'QUERY RESULTS 0'. A 'REFRESH' button is located in the top right corner.

CREACIÓN DE ELEMENTOS

Insert Document

```
1  _id : ObjectId("5bea5e3b1c9d44000028ff10 ")
2  matricula : "2123 "
3  nombre : "Juan "
4  apellido : "Lopez "
5  edad : 22
```

ObjectId

String

String

String

Int32

Cancel

Insert

CREACIÓN DE ELEMENTOS

The screenshot displays the MongoDB Compass interface. On the left sidebar, the database structure is shown: '2 DATABASES 3 COLLECTIONS'. Under 'db_test', the 'alumnos' collection is selected. The main panel shows the 'db_test.alumnos' collection with a 'COLLECTION SIZE 89B', 'TOTAL DOCUMENTS 1', and 'INDEXES TOTAL SIZE 4KB'. A search filter is applied: '{"filter": "example"}'. Below the filter, the query results are displayed as a single document:

```
{
  "_id": ObjectId("5bea5df9013411dad580243a"),
  "matricula": "2123",
  "nombre": "Juan",
  "apellido": "Lopez",
  "edad": 22
}
```

CONEXIONES

Connect to Cluster0

✓ Setup connection security

Choose a connection method

Connect

Choose a connection method [View documentation](#)

See methods to add data and diagnostics in the [Command Line Tools](#) shortcut from within your cluster.



Connect with the Mongo Shell

Mongo Shell with TLS/SSL support is required



Connect Your Application

Get a connection string and view driver connection examples



Connect with MongoDB Compass

Download Compass to explore, visualize, and manipulate your data



← Setup connection security

Close

EJEMPLOS

```
> show dbs
admin    0.000GB
config  0.000GB
local   0.000GB

> use uam
switched to db uam
```

EJEMPLOS

```
> db.alumnos.insert({matricula: 2123, nombre: "Carlos", edad: 19})
WriteResult({ "nInserted" : 1 })
> db.alumnos.insert({matricula: 2893, nombre: "Susana", edad: 23})
WriteResult({ "nInserted" : 1 })
> db.alumnos.insert({matricula: 2123, nombre: "Carlos", edad: 19})
WriteResult({ "nInserted" : 1 })
> db.alumnos.insert({matricula: 2893, nombre: "Susana", edad: 23})
WriteResult({ "nInserted" : 1 })
```

EJEMPLOS

```
> show collections
alumnos
materias
> db.alumnos.find().pretty()
{
  "_id" : ObjectId("5bea873bf134029b261ca8d9"),
  "matricula" : 2123,
  "nombre" : "Carlos",
  "edad" : 19
}
{
  "_id" : ObjectId("5bea8761f134029b261ca8da"),
  "matricula" : 2893,
  "nombre" : "Susana",
  "edad" : 23
}
>
```