

# Bases de Datos NoSQL

**DEV.FX**  
DESARROLLAMOS(PERSONAS);

dev

# Mongo DB

- Base de datos de documentos se clasifican bajo el término general NoSQL.
- Un registro en MongoDB es un documento, es una estructura de datos compuesta por pares de campo y valor.
- Los documentos MongoDB son similares a los objetos JSON.
- Los valores de los campos pueden incluir otros documentos, matrices y matrices de documentos.

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value  
← field: value  
← field: value  
← field: value

# Ventajas

- Los documentos corresponden a tipos de datos nativos en muchos lenguajes de programación.
- Los documentos y matrices integrados reducen la necesidad de uniones costosas.
- El esquema dinámico admite polimorfismo fluido.
  - El polimorfismo en MongoDB se refiere a la capacidad de una misma colección de documentos para contener diferentes tipos de documentos. En otras palabras, una colección puede contener documentos con diferentes estructuras y campos, lo que permite una mayor flexibilidad en el diseño de la base de datos y en el manejo de los datos.

# Ejemplo de polimorfismo



```
// Crear un documento de producto con un campo "nombre" común
```

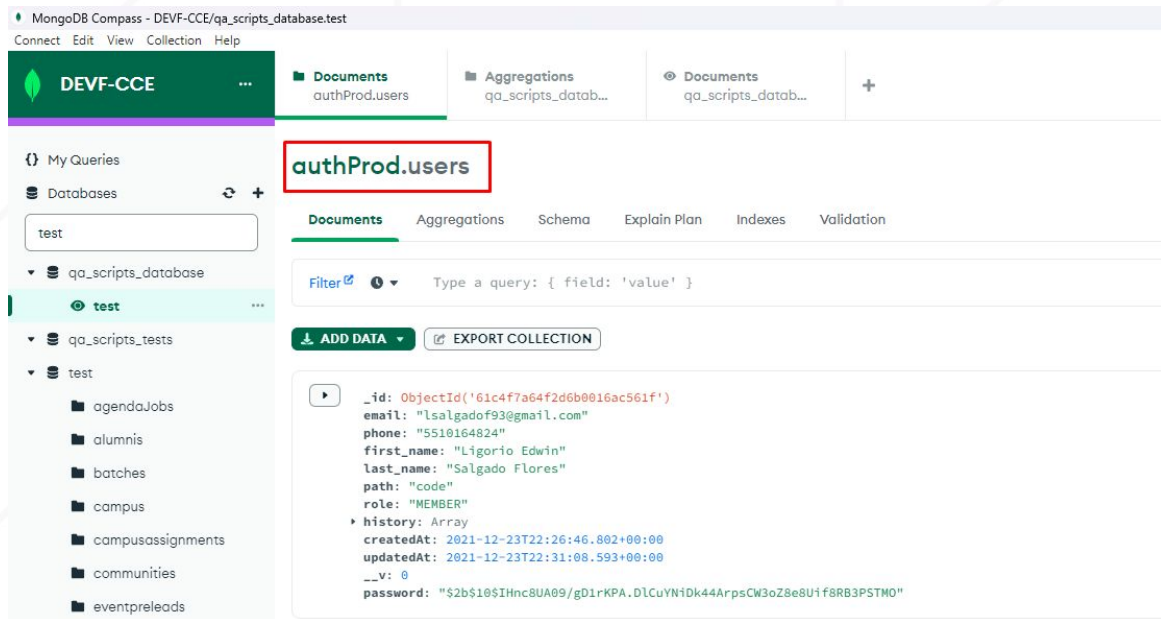
```
const producto1 = {  
  nombre: 'Producto 1',  
  precio: 10.0,  
  descripcion: 'Este es el producto 1',  
};
```

```
// Crear un documento de producto con un campo "marca" adicional
```

```
const producto2 = {  
  nombre: 'Producto 2',  
  precio: 20.0,  
  descripcion: 'Este es el producto 2',  
  marca: 'Marca 1',  
};
```

# Colecciones

En MongoDB, una colección es un conjunto de documentos almacenados en la base de datos MongoDB. Se podría pensar en una colección como el equivalente a una tabla en una base de datos relacional, pero sin una estructura de esquema fijo.



# Vistas

Son colecciones virtuales que no almacenan datos sino que proporcionan una vista personalizada de los datos en otra colección.

The screenshot displays the MongoDB Compass interface for a database named 'DEVF-CCE'. The left sidebar shows a tree view of databases and collections. The 'test' collection is selected under the 'qa\_scripts\_database' database. The main panel shows the 'Documents' tab for the 'qa\_scripts\_database.test' collection. A red box highlights the 'Documents' tab in the top navigation bar. Another red box highlights the 'READ-ONLY' and 'VIEW' buttons next to the collection name. The document content shows a JSON object with fields like '\_id', 'status', 'batch', 'start\_date', 'end\_date', 'product\_line', 'calendars', 'createdAt', 'updatedAt', and '\_\_v'.

MongoDB Compass - DEVF-CCE/qa\_scripts\_database.test

Connect Edit View Collection Help

DEVF-CCE

Documents authProd.users

Aggregations qa\_scripts\_datab...

Documents qa\_scripts\_datab...

qa\_scripts\_database.test READ-ONLY VIEW

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' }

EXPORT COLLECTION

```
{
  "_id": ObjectId("609eeb08f9f12f0017c7feb5"),
  "status": 1,
  "batch": 10,
  "start_date": "2021-06-28T00:00:00.000+00:00",
  "end_date": "2022-07-15T00:00:00.000+00:00",
  "product_line": "MASTER",
  "calendars": Object,
  "createdAt": "2021-05-14T21:26:32.909+00:00",
  "updatedAt": "2021-10-27T15:17:30.757+00:00",
  "__v": 3
}
```

# Características clave

- Persistencia de datos
- Alta disponibilidad
- Escalabilidad horizontal
- Compatibilidad con varios motores de almacenamiento

# Intro a mongoDB®


**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev



# Ejecutando Mongo

← → ↻ <https://cloud.mongodb.com/v2/5c5f3d655538556036558c8a#clusters> 🔑 ☆ ⓘ

 **mongoDB. Atlas** All Clusters ⚠ Please set your time zone Usage Thi

CONTEXT MATTI'S ORG - 2019-02-09 > PROJECT 0

Project 0 **Clusters**

PROJECT Clusters Overview Security


**Connect to Atlas**

Follow this checklist to get started.

0%

- ☐ Build your first cluster
- ☐ Create your first database user
- ☐ Whitelist your IP address
- ☐ Connect to your cluster


*No Thanks!*

 **Create a cluster**

Choose your cloud provider, region, and specs.

**Build a Cluster**

Once your cluster is up and running, live migrate an existing MongoDB database into Atlas with our [Live Migration Service](#).

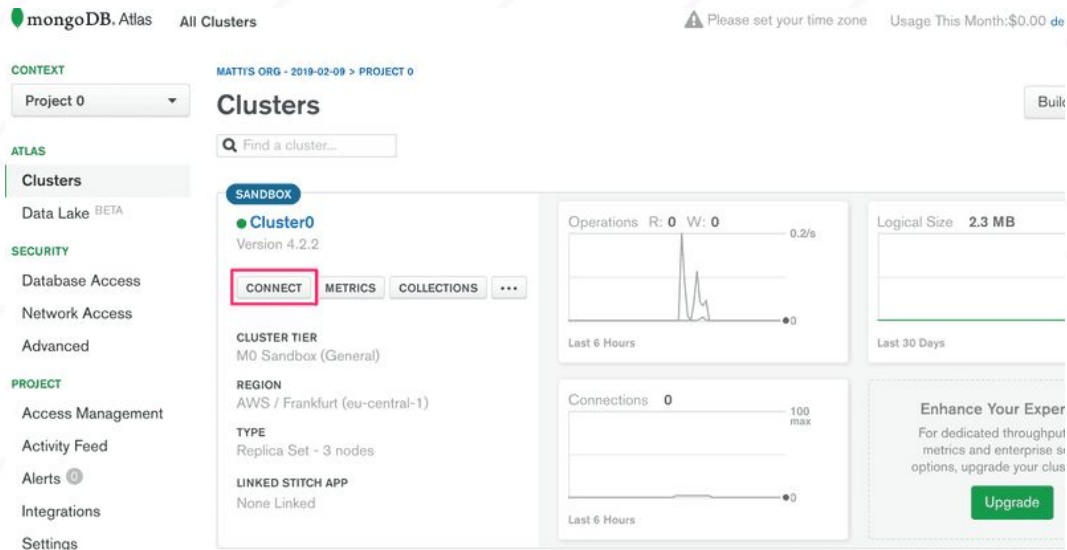
 **Get Started** <sup>4</sup>



**mongoDB**

# Pasos a seguir.

- Crear un clúster.
- Creamos un usuario y damos permisos
- Definir las direcciones IP
- Connect



The screenshot shows the MongoDB Atlas interface. At the top, it says 'mongoDB Atlas' and 'All Clusters'. There's a warning icon and text: 'Please set your time zone' and 'Usage This Month: \$0.00'. Below this, the 'CONTEXT' section shows 'Project 0' selected. The main heading is 'Clusters' with a search bar 'Find a cluster...'. On the left sidebar, under 'ATLAS', 'Clusters' is selected. Under 'SECURITY', 'Database Access', 'Network Access', and 'Advanced' are listed. Under 'PROJECT', 'Access Management', 'Activity Feed', 'Alerts', 'Integrations', and 'Settings' are listed. The main content area shows details for 'Cluster0' (Version 4.2.2). A red box highlights the 'CONNECT' button. Other buttons include 'METRICS', 'COLLECTIONS', and a three-dot menu. Below these are 'CLUSTER TIER' (M0 Sandbox (General)), 'REGION' (AWS / Frankfurt (eu-central-1)), 'TYPE' (Replica Set - 3 nodes), and 'LINKED STITCH APP' (None Linked). To the right of the cluster details are two charts: 'Operations' (R: 0, W: 0, 0.2/s) and 'Connections' (0, 100 max), both for the 'Last 6 Hours'. Further right, 'Logical Size' is 2.3 MB for the 'Last 30 Days'. At the bottom right, there's a section 'Enhance Your Experience' with a green 'Upgrade' button.

# A mover las manitas 🖐️

Controlador de MONGO.

- <https://mongodb.github.io/node-mongodb-native/>
- <https://mongoosejs.com/index.html>
- Iniciamos un proyecto vacío de node.js
  - `npm init -yes`
  - `npm i mongoose`

# Esquema de mongoose

Las bases de datos de documentos como Mongo no tienen esquema, lo que significa que la base de datos en sí no se preocupa por la estructura de los datos que se almacenan en la base de datos.

Es posible almacenar documentos con campos completamente diferentes en la misma colección.

La idea detrás de Mongoose es que los datos almacenados en la base de datos reciben un esquema al nivel de la aplicación que define la forma de los documentos almacenados en una colección determinada.

# MongoDB Compass

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

TOOLS

# Compass. The GUI for MongoDB.

Compass is a free interactive tool for querying, optimizing, and analyzing your MongoDB data. Get key insights, drag and drop to build pipelines, and more.

[Download Now](#)

[Read the docs](#) →

# Backend conectado a una base de datos

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# Esquema de mongoose

Las bases de datos de documentos como Mongo no tienen esquema, lo que significa que la base de datos en sí no se preocupa por la estructura de los datos que se almacenan en la base de datos.

Es posible almacenar documentos con campos completamente diferentes en la misma colección.

La idea detrás de Mongoose es que los datos almacenados en la base de datos reciben un esquema al nivel de la aplicación que define la forma de los documentos almacenados en una colección determinada.