

Introducción a MongoDB



En esta sesión daremos unas nociones básicas de cómo conectar y gestionar una base de datos MongoDB. Para los no iniciados en el tema, MongoDB es el principal representante, actualmente, de los sistemas de bases de datos NoSQL. Estos sistemas se han vuelto muy populares en los últimos años, y permiten dotar de persistencia a los datos de nuestra aplicación de una forma diferente a los tradicionales sistemas SQL.

En lugar de almacenar la información en tablas con sus correspondientes campos y registros, lo que haremos será almacenar estructuras de datos en formato BSON (similar a JSON), lo que facilita la integración con ciertas aplicaciones, como las aplicaciones Node.

1. Algunos conceptos de bases de datos NoSQL

Las bases de datos NoSQL y SQL comparten algunas similitudes, ya que ambas gestionan y almacenan información en bases de datos. Sin embargo, la principal diferencia entre ellas radica en **cómo organizan y almacenan los datos**.

En las bases de datos **SQL**, los datos se estructuran en **tablas** formadas por filas (o registros), y cada registro tiene una estructura fija, es decir, todos los registros en una tabla comparten los mismos **campos o columnas**.

Por otro lado, en las bases de datos **NoSQL**, como MongoDB, los datos se organizan en **colecciones**, que equivalen a las tablas de las bases de datos SQL. Cada colección contiene **documentos**, que son similares a los registros de SQL, pero con una diferencia clave: los documentos no tienen una estructura fija. Aunque es común que los documentos dentro de una misma colección compartan muchas de sus propiedades, cada uno puede tener un conjunto diferente de campos o propiedades, lo que proporciona una mayor flexibilidad para manejar datos no estructurados o semiestructurados.

En MongoDB, estos documentos se almacenan en formato **BSON** (una extensión binaria de JSON), lo que permite una fácil integración con lenguajes como JavaScript y una representación eficiente de tipos de datos complejos.

2. Instalación de MongoDB

MongoDB es, a día de hoy, el servidor de bases de datos No-SQL más utilizado y popular. Como hemos comentado, permite estructurar la información en colecciones, formadas por documentos, que se traducen de forma sencilla a objetos en JavaScript, de ahí su popularidad a la hora de cooperar con aplicaciones Node.js.

Versión requerida: se recomienda instalar la última versión disponible, ya sea mediante el gestor de paquetes de tu sistema operativo (como `apt` para Ubuntu o `brew` para macOS) o descargándola directamente desde la página oficial de MongoDB.

MongoDB se puede instalar y utilizar de tres maneras principales:

- **Instalación en local como una aplicación independiente:** permite ejecutar y detener MongoDB manualmente cuando sea necesario.
- **Instalación en local como servicio:** MongoDB se ejecuta en segundo plano y se inicia automáticamente con el sistema operativo, ideal para entornos de producción.
- **Uso de MongoDB en la nube:** a través de servicios como **MongoDB Atlas**, que proporciona una solución gestionada en la nube sin necesidad de instalarlo localmente.

Veremos aquí nociones de cómo instalarlo y ponerlo en marcha de las tres formas posibles.

2.1. Instalación como aplicación independiente

La instalación de MongoDB como aplicación independiente nos va a permitir ponerlo en marcha y detenerlo cuando nos convenga, utilizando comandos específicos o scripts personalizados.

Para descargar e instalar MongoDB, debes dirigirte a su sitio [web oficial](#) y, en el menú superior, seleccionar la sección de **Productos**. Allí, encontrarás la opción de descargar la versión **Community Server**, que es gratuita y no requiere registro previo ([enlace directo a la descarga](#)). Elige la versión adecuada para tu sistema operativo.

2.1.1. Sistemas Mac y Linux

En el caso de **Mac OSX** o **Linux**, puedes descargar un archivo TGZ, y descomprimirlo en la carpeta que quieras. Normalmente se suele ubicar en una carpeta llamada `mongo` en la carpeta personal del usuario (por ejemplo, `/home/alumno/mongo`).

En el caso de Ubuntu y distribuciones basadas en Debian, es más conveniente descargar el `.deb` e instalarlo desde consola con el comando `$ sudo dpkg -i nombrePaquete.deb`.

En **Ubuntu 20.04 y versiones más recientes**, no es necesario instalar la librería **libssl1.1**, ya que las versiones actuales de MongoDB dependen de **libssl3**. Sigue las [instrucciones oficiales en la documentación de MongoDB](#) para una correcta instalación.

Si estás utilizando una versión anterior de Ubuntu que aún requiere **libssl1.1**, puedes añadir el repositorio `focal-security` ejecutando este comando en una sola línea::

```
$ echo "deb http://security.ubuntu.com/ubuntu focal-security main" |  
sudo tee /etc/apt/sources.list.d/focal-security.list
```

Finalmente, actualiza los repositorios: `$ sudo apt-get update`

2.1.2. Sistemas Windows

En el caso de **Windows**, puedes descargar un archivo ZIP y proceder como en el caso anterior (descomprimirlo donde quieras), o bien utilizar el instalador MSI y seguir los pasos del asistente (elegir instalación *Completa*). Conviene recordar la carpeta en la que se instala MongoDB (`C:\Archivos de programa\MongoDB\Server\X.Y`, siendo X.Y la versión descargada), ya que habrá que acceder a esta carpeta para arrancar el servidor manualmente.

2.1.3. La carpeta para las bases de datos

Una vez que hayas instalado o descomprimido MongoDB en tu sistema, es necesario crear una carpeta donde se almacenarán las bases de datos. Aunque puedes llamarla como prefieras, es común nombrarla **mongo-data**. Esta carpeta debe ubicarse en un lugar accesible y fácil de localizar.

- En **Linux** o **MacOS**, puedes crear la carpeta en el directorio personal del usuario, por ejemplo: `/home/usuario/mongo-data`.
- En **Windows**, puedes crearla en una ubicación como `C:\mongo-data`.

Es importante asegurarse de que MongoDB tenga los permisos adecuados para escribir en esta carpeta, especialmente en sistemas basados en Unix.

2.1.4. Puesta en marcha del servidor

Para iniciar el servidor MongoDB en cualquier sistema, debes acceder desde un terminal a la subcarpeta `bin` de la carpeta de instalación de MongoDB, y ejecutar desde el propio terminal el comando `mongod`, indicando en el parámetro `--dbpath` la ruta hacia la carpeta que hemos creado para almacenar los datos.

- Ejemplo en **Mac OS X** o **Linux**:

```
./mongod --dbpath /home/alumno/mongo-data
```

NOTA: Si no sabes dónde se ha instalado *mongo*, usa el comando `$ whereis mongod`

- Ejemplo en **Windows**: abre el *Símbolo del sistema* (cmd) o *PowerShell* y ejecuta el comando `mongod.exe` con los mismos parámetros.

```
mongod.exe --dbpath C:\mongo-data
```

En cualquier caso, se mostrarán unos cuantos mensajes por la consola, y uno de ellos indicará que MongoDB queda a la espera de conexiones en el puerto por defecto, el 27017:

```
2022-10-11T01:12:27.195+0100 I NETWORK
[thread1] waiting for connections on port 27017
```

2.2. Instalación como servicio

La instalación de MongoDB como servicio deja el servidor instalado y puesto en funcionamiento en el sistema, y proporciona una serie de comandos para poder detener y reanudar el servicio desde terminal. Esta opción es particularmente útil en sistemas de producción (no tanto de pruebas o desarrollo) para dejar la base de datos permanentemente en marcha.

Para dejar instalado MongoDB como servicio en sistemas Linux (algo muy habitual), podemos seguir los pasos indicados en la [web oficial](#), para la versión de MongoDB deseada (se puede elegir en un desplegable en la sección izquierda).

Una vez instalado podemos utilizar estos comandos para conocer el estado actual del servidor, pararlo, iniciarlo o reiniciarlo, respectivamente:

```
sudo systemctl status mongod
sudo systemctl stop mongod
sudo systemctl start mongod
sudo systemctl restart mongod
```

Para permitir que MongoDB se inicie automáticamente en cada reinicio del sistema, ejecutamos este comando:

```
sudo systemctl enable mongod
```

Instalación en otros sistemas

Si prefieres instalar MongoDB como servicio en otros sistemas, aquí tienes algunas guías para hacerlo, ya que estos contenidos no forman parte del curso en sí:

- [Windows](#)
- [MacOSX](#)

2.3. Uso de MongoDB en la nube

En el caso de que no queramos (o no haya sido posible) instalar MongoDB en nuestro sistema, podemos utilizar las bases de datos MongoDB en la nube proporcionada a través de *MongoDB Atlas*. En [esta web](#) podemos registrarnos y elegir el plan gratuito, que ofrece una capacidad máxima de 512MB de almacenamiento, pero suficiente para lo que vamos a probar durante este curso.

MongoDB.

MONGODB ATLAS

MongoDB Atlas.

La plataforma de datos para desarrolladores de multicloud.

Un conjunto integrado de base de datos de cloud y servicios de datos para acelerar y simplificar la forma de construir con datos.

[Proba Gratis](#) [Contacta el departamento de ventas](#) →

Cluster

Serverless

Connect To Your Database →

Una vez nos hemos identificado, seleccionamos en el menú de la izquierda **Deployment > Database**, y a continuación creamos un **cluster**.

Atlas May's ... Access Manager Billing All Clusters Get Help M

Proje... Data Services Charts

Overview MAY'S ORG - 2023-10-02 > PROJECT 0

Clusters

Database

Services

Atlas Search

Stream Processing

Triggers

Migration

Data Federation

Programmatic Access

Data API

SECURITY

Quickstart

Backup

Database Access

Create a cluster

Choose your cloud provider, region, and specs.

[Build a Cluster](#)

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

Al crear el cluster, podemos elegir la plataforma cloud donde se desplegará la base de datos (Amazon, Azure, Google) y la ubicación. La elección de una u otra opción no es importante para lo que haremos en el curso, aunque cuanto más cerca elijamos la ubicación más rápida será la conexión a la base de datos. También tendremos que elegir el nombre del *cluster* que se creará (no es muy relevante, pero sí conviene tenerlo presente porque dentro definiremos las bases de datos que necesitamos).

Deploy your cluster

Use a template below or set up advanced configuration options. You can also edit these configuration options once the cluster is created.

M10 \$0.09/hour

For production applications with sophisticated workload requirements.

STORAGE	RAM	vCPU
10 GB	2 GB	2 vCPUs

Serverless

For application development and testing, or workloads with variable traffic.

STORAGE	RAM	vCPU
Up to 1 TB	Auto-scale	Auto-scale

M0 Free

For learning and exploring MongoDB in a cloud environment.

STORAGE	RAM	vCPU
512 MB	Shared	Shared

✔ **Free forever!** Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Name
You cannot change the name once the cluster is created.

Automate security setup ⓘ
 Preload sample dataset ⓘ

Provider

aws

Google Cloud

Azure

Region

Paris (eu-west-3) ★ 🌱

★ Recommended ⓘ 🌱 Low carbon emissions ⓘ

Después de que el clúster esté creado, es necesario configurar quién puede acceder a la base de datos. Seleccionamos desde el menú de la izquierda **Security > Quickstart**. Podemos hacerlo a través de usuario y contraseña o a través de certificado; en nuestro caso elegiremos la primera opción, y pulsamos el botón para generar el usuario (*Create User*).

Atlas May's Org - ... Access Manager Billing All Clusters Get Help May

Project 0 Data Services Charts

Overview
DEPLOYMENT
Database
SERVICES
Atlas Search
Stream Processing
Triggers
Migration
Data Federation
Programmatic Access
Data API
SECURITY
Quickstart
Backup
Database Access
Network Access
Advanced
New On Atlas 4
Goto

MAY'S ORG - 2023-10-02 > PROJECT 0

Security Quickstart

To access data stored in Atlas, you'll need to create users and set up network security controls. [Learn more about security setup](#)

✔ How would you like to authenticate your connection?

Your first user will have permission to read and write any data in your project.

Username and Password

Certificate

Create a database user using a username and password. Users will be given the *read and write to any database privilege* by default. You can update these permissions and/or create additional users later. Ensure these credentials are different to your MongoDB Cloud username and password. You can manage existing users via the [Database Access Page](#).

Username

Password ⓘ

Autogenerate Secure Password
Copy

Create User

Un poco más abajo generamos las ubicaciones permitidas para acceder a la base de datos. Podemos elegir a mano algunas direcciones IP predeterminadas, o pulsar el botón de *Add my current IP address* para que se genere automáticamente la IP desde la que nos conectamos actualmente.

✓ Where would you like to connect from?

Enable access for any network(s) that need to read and write data to your cluster.

My Local Environment

Use this to add network IP addresses to the IP Access List. This can be modified at any time.

Cloud Environment

Use this to configure network access between Atlas and your cloud or on-premise environment. Specifically, set up IP Access Lists, Network Peering, and Private Endpoints.

Add entries to your IP Access List

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the [Network Access Page](#).

IP Address	Description	
<input type="text" value="Enter IP Address"/>	<input type="text" value="Enter description"/>	<input type="button" value="Add My Current IP Address"/>
<input type="button" value="Add Entry"/>		
IP Access List	Description	
88.19.21.109/32		<input type="button" value="EDIT"/> <input type="button" value="REMOVE"/>
<input style="border: 2px solid black;" type="text" value=""/>	My IP Address	<input type="button" value="EDIT"/> <input type="button" value="REMOVE"/>
213.0.87.202/32		<input type="button" value="EDIT"/> <input type="button" value="REMOVE"/>
88.19.23.213/32	Created as part of the Auto Setup process	<input type="button" value="EDIT"/> <input type="button" value="REMOVE"/>

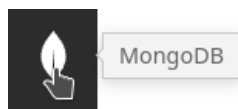
3. Herramientas de conexión con el servidor

Para poder conectar al servidor y gestionar las bases de datos que tengamos instaladas podemos utilizar diversas herramientas. En este apartado propondremos dos de ellas:

- Un *plugin* o extensión que podemos instalar y utilizar directamente desde Visual Studio Code
- Una herramienta gratuita de escritorio, como *Compass* o *Studio3T*

3.1. Plugin MongoDB para VS Code

Como primera opción podemos instalar y utilizar un plugin como *MongoDB for vscode*, que podemos consultar en el [marketplace oficial](#). Una vez instalado, desde la paleta de herramientas de VS Code tendremos acceso directo a MongoDB:



Aquí debemos configurar la conexión a nuestro servidor MongoDB (suponiendo que lo tengamos iniciado ya, como se ha explicado anteriormente). Pulsamos en el enlace *Add Connection* de la parte superior izquierda y configuramos los parámetros de conexión (también podemos editar conexiones o borrarlas desde estos enlaces):

New connection ✕

General SSL/TLS SSH Tunnel Advanced

Connection Type

Standalone Replica Set SRV Record

Hostname localhost Port 27017 +

Authentication

None Username / Password SCRAM-SHA-256 LDAP

X.509

Cancel Connect

Deberemos verificar la dirección y puerto, que por defecto son *localhost* y *27017* respectivamente. Nos conectamos a la base de datos, pulsando en el botón *Connect*. Tras conectar, en el panel izquierdo tendremos el explorador para examinar los elementos del servidor Mongo. Cuando creamos una base de datos, aparecerá en el listado, junto con las tablas o colecciones que contiene. Haciendo clic en una colección, podemos ver a la derecha los documentos que contiene.

En el caso de que queramos conectar a un *cluster* de MongoDB Atlas, simplemente especificamos la URL de conexión junto con los parámetros de autenticación correspondientes. Podemos acceder a nuestra cuenta *Atlas* y pulsar en el botón *Connect* del *cluster* al que queramos conectar para obtener la URL de conexión:

Connect to Cluster0

Set up connection security ✓ Choose a connection method ✓ **3 Connect**

Connecting with MongoDB Driver

1. Select your driver and version
We recommend installing and using the latest driver version.

Driver: Node.js Version: 5.5 or later

2. Install your driver
Run the following on the command line

```
npm install mongodb
```

[View MongoDB Node.js Driver installation instructions.](#)

3. Add your connection string into your application code

Use this connection string in your application

View full code sample

```
mongodb+srv://maycal2425:<db_password>@cluster0.epx1o.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
```

Replace **<db_password>** with the password for the **maycal2425** database user. Ensure any option params are [URL encoded](#).

Desde VS Code, creamos una conexión como *Connection String* y pegamos la URL que nos indique Atlas (incluyendo nuestro password en ella):

mongodb+srv://nachoiborra:<password>@cluster0.bmlrymc.mongodb.net/

Enter your connection string (SRV or standard) (Press 'Enter' to confirm or 'Escape' to cancel)

MongoDB

Navigate your databases and collections, use playgrounds for exploring data

Connected to: **localhost:27017**

All set. Ready to start?
Create a playground.

Connect with **Connection String**

Connect

Advanced **Connection**

Open for

Ctrl + Shift + P for all MongoDB Command Pal

3.1.1 Probando MongoDB en el playground

MongoDB Playground es una funcionalidad disponible en el editor de **Visual Studio Code** que permite ejecutar comandos de MongoDB directamente desde un archivo de texto sin necesidad de crear un servidor o usar Node.js. Es una herramienta útil para realizar pruebas rápidas y ejecutar comandos de MongoDB de forma interactiva.

Para probarlo, **conéctate a tu instancia de MongoDB** y haz clic en el botón **New Playground**. Esto abrirá un archivo de texto donde podrás escribir y ejecutar comandos de MongoDB.

Escribe el siguiente código de ejemplo:

```
const database = "DAW";
const collection = "Contactos";

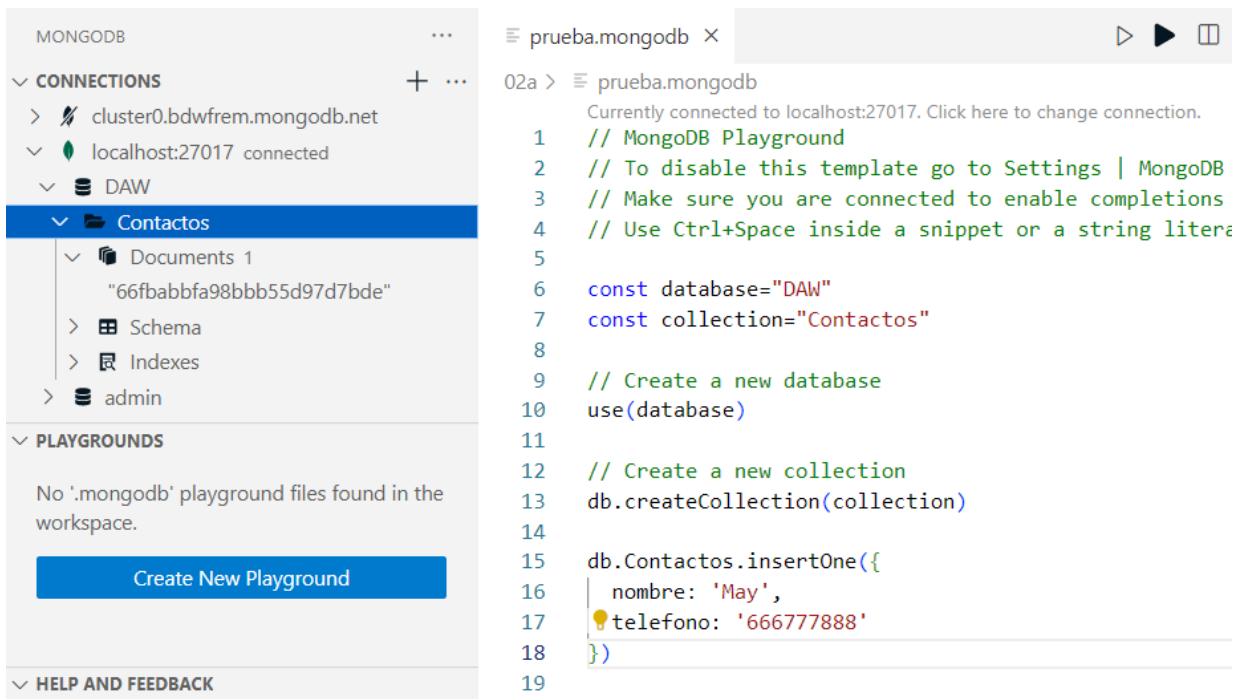
// Crear una nueva base de datos
use(database);

// Crear una nueva colección
db.createCollection(collection);

// Insertar un documento en la colección
db.Contactos.insertOne({
  nombre: 'May',
  telefono: '666777888'
});
```

Observa que no es necesario crear la variable `db`. El entorno del playground ya define `db` por ti. Cuando haces `use(database)`, MongoDB automáticamente cambia el contexto de `db` a la base de datos especificada en el comando `use`.

Haz clic en el botón **Play** (o en "Run All" en la parte superior del archivo del Playground) para ejecutar el código. Verás la salida del código en el panel de salida, que te mostrará el resultado de la inserción del documento.



The screenshot shows the MongoDB Playground interface. On the left, there's a sidebar with 'CONNECTIONS' (cluster0.bdwm.mongodb.net, localhost:27017 connected, DAW, Contactos) and 'PLAYGROUNDS' (no files found). The main area shows a code editor with the following code:

```

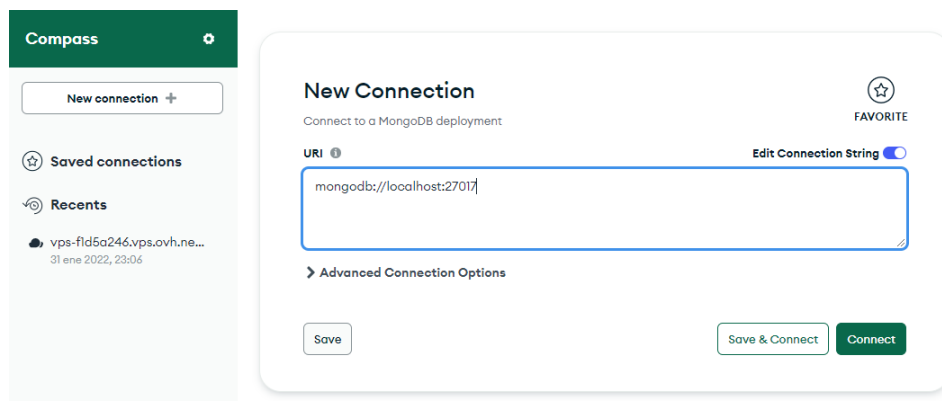
02a > prueba.mongodb
Currently connected to localhost:27017. Click here to change connection.
1 // MongoDB Playground
2 // To disable this template go to Settings | MongoDB
3 // Make sure you are connected to enable completions
4 // Use Ctrl+Space inside a snippet or a string literal
5
6 const database="DAW"
7 const collection="Contactos"
8
9 // Create a new database
10 use(database)
11
12 // Create a new collection
13 db.createCollection(collection)
14
15 db.Contactos.insertOne({
16   nombre: 'May',
17   telefono: '666777888'
18 })
19

```

3.2. Utilizando Compass

Compass es una herramienta de gestión de bases de datos MongoDB gratuita, y disponible en la propia web de MongoDB ([aquí](#) el enlace).

Desde la herramienta tendremos un pequeño formulario de conexión, que podemos utilizar tanto para conectarnos a una base de datos local...



The screenshot shows the MongoDB Compass interface. On the left, there's a sidebar with 'Compass', 'New connection +', 'Saved connections', and 'Recents'. The main area shows the 'New Connection' dialog box with the following content:

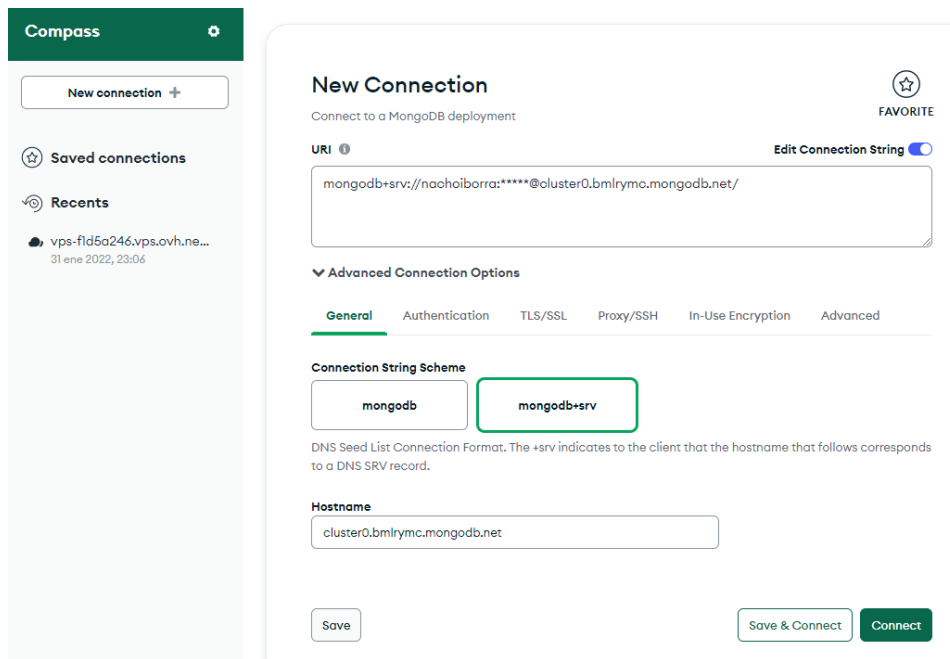
New Connection
Connect to a MongoDB deployment

URI Edit Connection String

Advanced Connection Options

Save Save & Connect Connect

... como a una base de datos en la nube, proporcionando la URL que nos indique Atlas:



3.3. Utilizando *Studio3T*

La herramienta *Studio3T* es una evolución de una herramienta previa llamada *Robomongo*, que se empleaba en versiones anteriores para conectar con MongoDB. Esta herramienta posteriormente evolucionó en *Robo3T*, que era una versión gratuita de otra comercial llamada *Studio3T*. Con el paso de los años, todo se ha unificado con el nombre *Studio3T*, que ofrece la herramienta tanto de forma comercial como gratuita, a través de [este enlace](#).

En estos apuntes no explicaremos cómo utilizarla, ya que su uso es muy similar a *Compass*. Pero dejamos esta opción también para quienes no os termine de convencer ninguna de las alternativas propuestas.

3.4. Resumen

[Aquí](#) tienes disponible un vídeo resumen de cómo instalar MongoDB como aplicación independiente en Windows/Mac/Linux, y de cómo usar MongoDB Atlas desde VS Code o desde Compass.