

GRUPO C2 ESPAÑITA

- Rafael Calvo Córdoba
- Jose María Ramirez González
- Alejandro Herrera Dadgar
- Alberto Llamas González

Índice

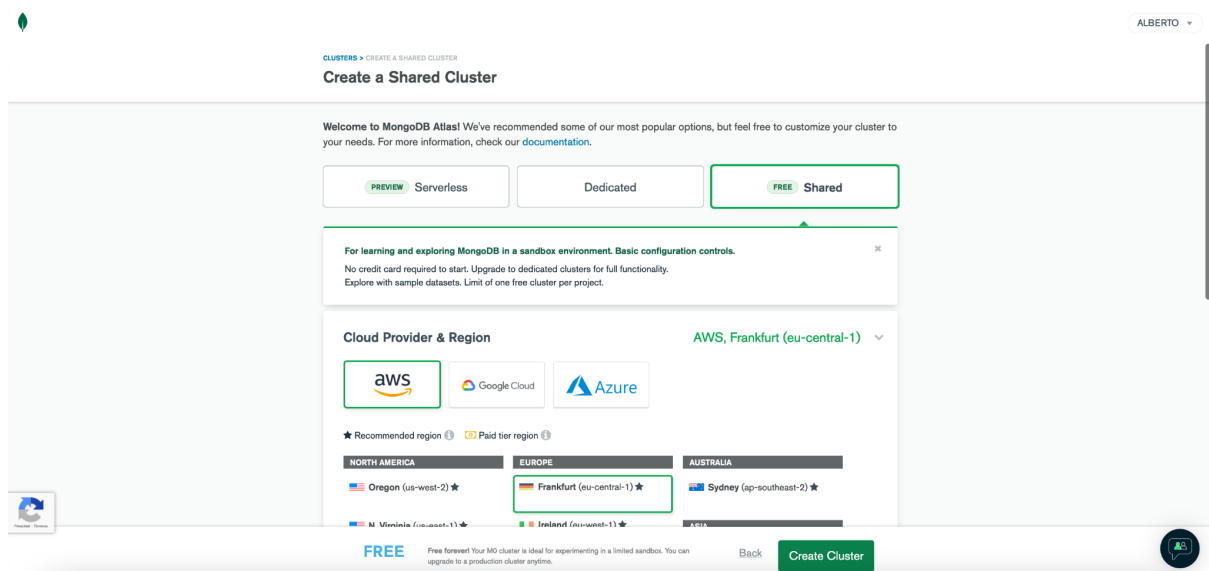
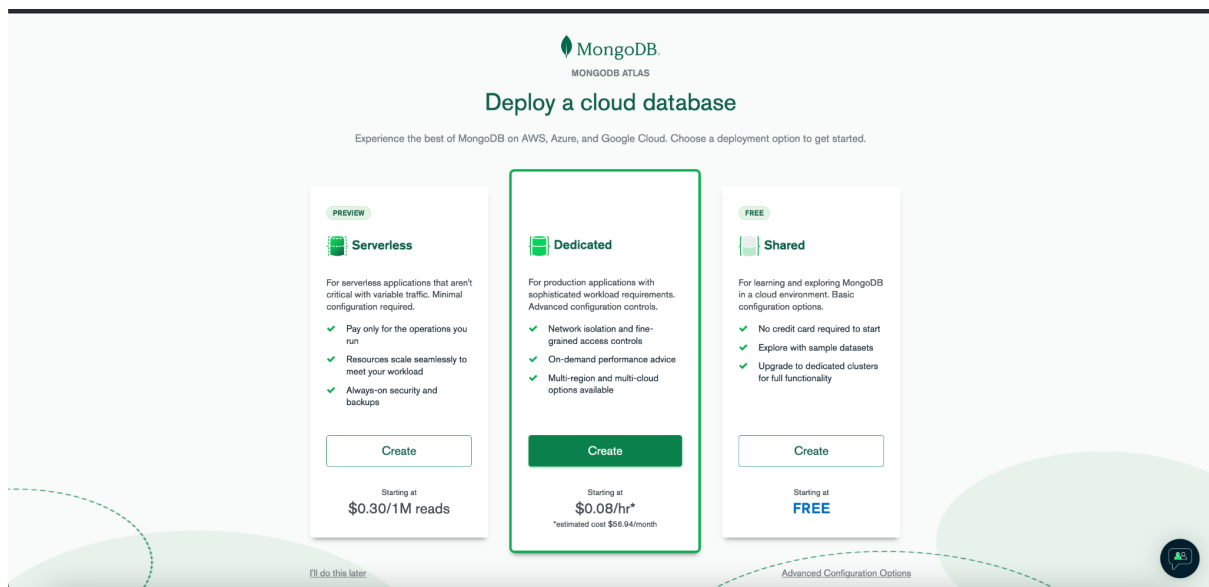
Descripción de la descarga e instalación de MongoDB	2
Descripción de DDL y DML	5
Creación de estructuras, inserción/modificación/borrado de datos, y consultas	7
Mecanismo de conexión a MongoDB	10
¿Sería adecuado para implementar el SI del torneo de tenis?	12
Bibliografía	12

1. Descripción de la descarga e instalación de MongoDB

Para la realización de este trabajo, hemos decidido escoger el lenguaje NoSQL **MongoDB**. Hemos creado tanto una base de datos en nuestro localhost, como en el servicio Atlas(clúster en la nube) que ofrece MongoDB.

Instalación y configuración de servicio Atlas y MongoDB Compass

En primer lugar, tenemos que registrarnos y seleccionar la opción gratis Shared. Después creamos dicho cluster compartido en un servidor Amazon (AWS, Amazon Web Service) en Frankfurt (por escoger un lugar). Una vez elegida la configuración, creamos nuestro cluster.



Una vez creado y desplegado el cluster, tenemos que conectarnos. Usaremos MongoDB Compass el cual se puede descargar de forma gratuita [aquí](#). También creamos un usuario, al que daremos permisos de administrador y añadiremos nuestra IP para poder conectarnos.

ALBERTO'S ORG - 2021-12-26 > PROJECT 0

Database Access

Database Users

Custom Roles

+ ADD NEW DATABASE USER

User Name	Authentication Method	MongoDB Roles	Resources	Actions
albertollamas	SCRAM	readWriteAnyDatabase@admin	All Resources	<div>EDITDELETE</div>

ALBERTO'S ORG - 2021-12-26 > PROJECT 0

Network Access

IP Access List

Peering

Private Endpoint

+ ADD IP ADDRESS

You will only be able to connect to your cluster from the following list of IP Addresses:

IP Address	Comment	Status	Actions
(includes your current IP address)	My IP Address	Active	<div>EDITDELETE</div>

Para conectarnos al cluster nos vamos a Connect donde nos aparece lo siguiente para que nos descarguemos MongoDB Compass. Una vez instalado copiamos dicho enlace del punto (2) de la imagen en MongoDB Compass.

ALBERTO'S Org - 20...

Access Manager

Billing

Project 0

Atlas

Realm

DEPLOYMENT

Databases

Data Lake

DATA SERVICES

Triggers

Data API

PREVIEW

SECURITY

Database Access

Network Access

Advanced

Cluster0

Connect

View Monitor

VERSION

4.4.10

REGION

AWS / Frankfurt (eu-central-1)

Get Started

System Status: All Good

©2021 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Connect to Cluster0

Setup connection security

Choose a connection method

Connect

I do not have MongoDB Compass

I have MongoDB Compass

1

Select your operating system and download MongoDB Compass

OS X 64-bit (10.10+)

Download Compass (1.29.6)

Copy download URL

2

Copy the connection string, then open MongoDB Compass.

mongodb+srv://albertollamas:password@cluster0.eervh.mongodb.net/test

You will be prompted for the password for the albertollamas user's (Database User) username. When entering your password, make sure that any special characters are URL encoded.

Having trouble connecting? View our troubleshooting documentation

Go Back

Close

All Clusters

Get Help

ALBERTO

Create

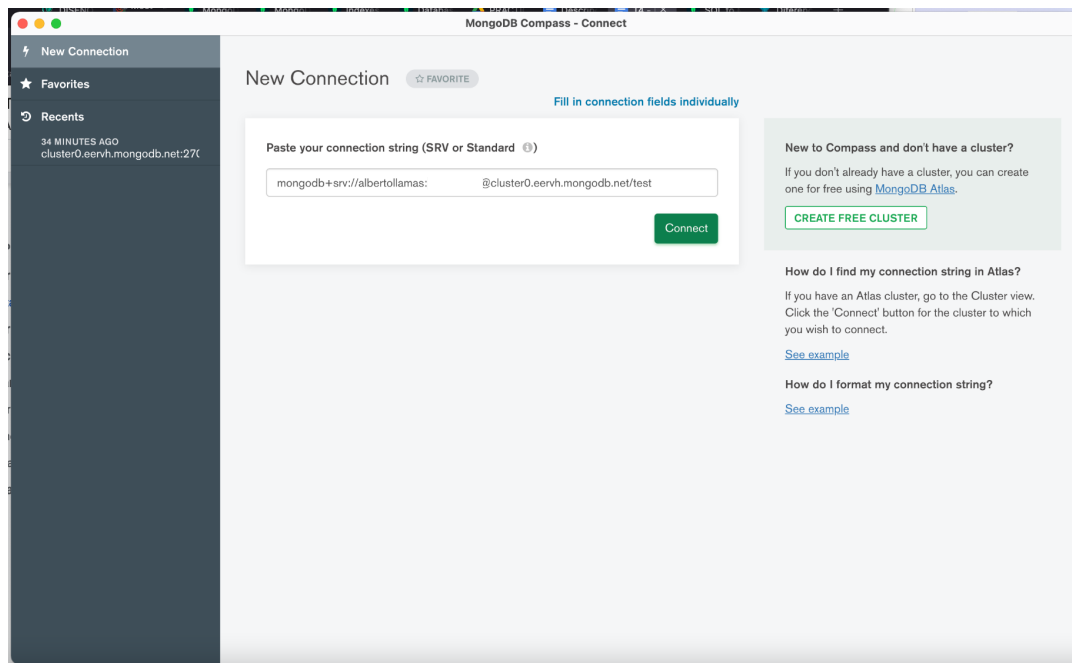
FREE

SHARED

Enhance Your Experience

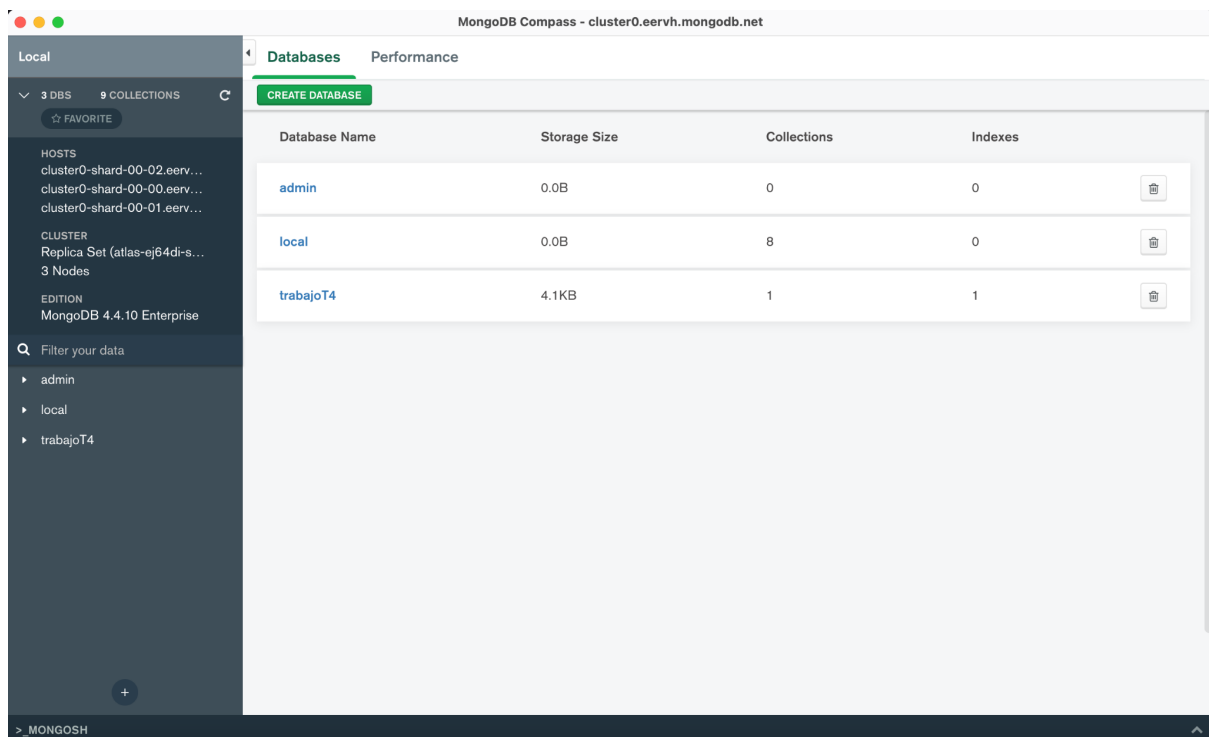
For production throughput and richer metrics, upgrade to a dedicated cluster now!

Upgrade



(Por motivos de seguridad, hemos ocultado la contraseña)

Y ya estaríamos conectados al cluster creado.



2. Descripción de DDL y DML

DDL

- Sentencia equivalente a **CREATE**: `db.createCollection(nombre, opción)`.
Esta sentencia crea una Colección, el equivalente de una tabla en SQL.
Toma un nombre y una serie de opciones como parámetros.
Algunas de las diferentes opciones que nos interesan son:
 - *capped*, para establecer un tamaño máximo de colección. Es necesario especificar el tamaño.
 - *validator*, para permitir a los usuarios establecer las reglas de verificación de valores, proporcionando un documento con las nuevas reglas.
 - *viewOn*, para especificar el origen desde el que se ha de mostrar la tabla.
- No existe una sentencia equivalente a la sentencia **ALTER** de SQL, ya que Mongo no fuerza un estado concreto para los elementos de una colección, por lo que no podemos modificar toda una colección como en SQL.
Aún así, a nivel de documento podemos usar `updateMany()` con el operador `$set` y `$unset` para añadir y eliminar “columnas” respectivamente, sobre cada documento en la colección, siendo cada documento el equivalente de una fila.
El uso de `updateMany` es como sigue: `db.NombreColeccion.updateMany({ }, { $set/$unset : {Sentencia} })`.
- La sentencia equivalente a **DROP** es, para sorpresa de todos, **DROP**.
Esta se usa de la siguiente manera: `db.NombreColeccion.drop`

DML

- Respecto a la sentencia **INSERT**, tenemos 2 alternativas diferentes.
 - Si queremos añadir un solo ítem, usamos `db.NombreColeccion.insertOne()`.
 - Si queremos añadir más de uno, usamos `db.NombreColeccion.insertMany()`.Ambas requieren que especifiquemos el/los ítem(s) a insertar.
- La sentencia **SELECT** se escribe como `find`, usándose de la siguiente manera:
`db.NombreColeccion.find({Condición}, {Valores a mostrar})`.
Se recomienda mirar más información, que puede encontrar en [este enlace](#), para comprender cómo realizar un uso profesional y adecuado de esta sentencia.
- Respecto a **UPDATE**, se usa la sentencia que usamos con ALTER ya que, como comentamos, ALTER no tiene sentencia equivalente, si no que actúa a nivel individual sobre cada documento.
Así pues, el uso de *updateMany* es el que sigue:
`db.NombreColeccion.updateMany({ Condición }, { Actualización })`
Importante resaltar de nuevo el uso de `$set` en la zona de *actualización* de *updateMany* para ejecutar correctamente algunas sentencias.
- La sentencia **DELETE** se puede ejecutar en MongoDB haciendo uso de *deleteMany*.
Este comando se usa de la siguiente manera:
`db.NombreColeccion.deleteMany({ Condición })`
Lo que se consigue con esta sentencia es borrar aquellos documentos donde se cumpla la condición.

3. Creación de estructuras, inserción/modificación/borrado de datos, y consultas

Para la creación de estructuras, inserción, modificación, borrado de datos y consultas hemos utilizado la aplicación MongoDB Compass. Los comandos para las sentencias, se han descrito en el punto anterior.

DDL

Create:

Create Collection

Collection Name

Partidos

☐ Capped Collection

Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. ⓘ

☐ Use Custom Collation

Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. ⓘ

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time.

Cancel

Create Collection

Drop:

Partidos	3	74.0 B	222.0 B	1	36.9 KB
----------	---	--------	---------	---	---------



Drop Collection

⚠ To drop **Partidos** type the collection name **Partidos**.

Partidos

Cancel

Drop Collection

DML

Insert:

Insert to Collection seminario3.Tenista

VIEW

{}

≡

1

▼

/**

2

*

Paste one or more documents here

3

*/

4

{

5

▼

6

7

8

9

10

}

Cancel

Insert

Insert to Collection seminario3.Arbitros

VIEW

{}

≡

1

▼

/**

2

*

Paste one or more documents here

3

*/

4

{

5

▼

6

7

8

9

10

}

Cancel

Insert

Insert to Collection seminario3.Partidos

VIEW

{}

≡

1

▼

/**

2

*

Paste one or more documents here

3

*/

4

{

5

▼

6

7

8

9

10

▼

11

12

13

}

Cancel

Insert

Select:

FILTER

{ field: 'value' }

OPTIONS

FIND

RESET

ADD DATA

VIEW

Displaying documents 1 - 3 of 3

REFRESH

_id: ObjectId("61c8a316c25689ee0e832a13")

id_tenista: "18"

id_arbitro: "3"

fecha: 2021-12-26T23:00:00.000+00:00

_id: ObjectId("61c8a398c25689ee0e832a14")

id_tenista: "11"

id_arbitro: "4"

fecha: 2021-12-29T23:00:00.000+00:00

_id: ObjectId("61c8a3aec25689ee0e832a15")

id_tenista: "11"

id_arbitro: "7"

fecha: 2021-12-29T23:00:00.000+00:00

FILTER

{id_arbitro:"4"}

OPTIONS

FIND

RESET

ADD DATA

VIEW

Displaying documents 1 - 1 of 1

REFRESH

_id: ObjectId("61c8a398c25689ee0e832a14")

id_tenista: "11"

id_arbitro: "4"

fecha: 2021-12-29T23:00:00.000+00:00

Update:

1

_id: ObjectId("61c8a26bc25689ee0e832a12")

ObjectId

2

id_tenista: "19"

String

3

id_arbitro: "3"

String

4

fecha: 2021-12-26T23:00:00.000+00:00

Date

Document Modified.

CANCEL

UPDATE

Delete:

_id: ObjectId("61c8a26bc25689ee0e832a12")

id_tenista: "19"

id_arbitro: "3"

fecha: 2021-12-26T23:00:00.000+00:00

Document Flagged For Deletion.

CANCEL

DELETE

4. Mecanismo de conexión a MongoDB

Hemos decidido conectarnos a la Base de Datos a través de Python.

Prerrequisitos.

Como prerrequisito necesitamos descargar e instalar Python en nuestra máquina. Podemos obtenerlo a través de su página oficial, usando el siguiente enlace:

<https://www.python.org/downloads/>

Conectando Python y MongoDB Atlas.

Para ello necesitamos hostear una base de datos de MongoDB de forma gratuita usando MongoDB Atlas(<https://www.mongodb.com/es/cloud/atlas/register>) .

Necesitamos instalar PyMongo que tiene un conjunto de paquetes para la interacción Python MongoDB.

Para instalar PyMongo abrimos la línea de comandos y escribimos:

“python -m pip install pymongo”.

Si se recibe el error “ModuleNotFoundError: No module named 'pymongo'”, hay que desinstalar pymongo usando el comando: “pip uninstall pymongo”. Tras esto se vuelve a instalar PyMongo.

Para realizar la conexión Python-MongoDB usamos mongodb srv URI, por lo que debemos instalar dnspython. Usamos el siguiente comando: “python -m pip install dnspython”.

Ahora, podemos usar pymongo como una biblioteca de python mongodb en nuestro código con una declaración de importación.

Creando una base de datos MongoDB en Python

El primer paso para conectar Python a Atlas es la configuración del clúster MongoDB(<https://docs.atlas.mongodb.com/tutorial/create-new-cluster/>) .

A continuación, creamos un archivo con el nombre “pymongo_test_insert.py” en cualquier carpeta para escribir código pymongo. Podemos utilizar cualquier editor de texto simple.

Creamos el cliente mongodb agregando lo siguiente:

```
def get_database():
    from pymongo import MongoClient
    import pymongo

    # Provide the mongodb atlas url to connect python to mongodb using pymongo
    CONNECTION_STRING = "mongodb+srv://<username>:<password>@<cluster-name>.mongodb.net/myFirstDatabase"

    # Create a connection using MongoClient. You can import MongoClient or use pymongo.MongoClient
    from pymongo import MongoClient
    client = MongoClient(CONNECTION_STRING)

    # Create the database for our example (we will use the same database throughout the tutorial
    return client['user_shopping_list']

# This is added so that many files can reuse the function get_database()
if __name__ == "__main__":

    # Get the database
    dbname = get_database()
```

Utilizamos “connection_string” para crear el mongoclient y obtener la conexión a la base de datos MongoDB. Cambiamos el nombre de usuario, la contraseña y el nombre del clúster. MongoDB no crea una base de datos hasta que tenga colecciones y documentos en ella.

5. ¿Sería adecuado para implementar el SI del torneo de tenis?

Para implementar el SI que se nos pide, sería posible usando una base de datos no relacional, pero complicaría los modelos más de lo necesario

Esto se debe a que nuestro SI, por definición, son un conjunto de **relaciones**, en las que los elementos están interconectados y siguen una estructura definida.

Una base de datos no relacional se usa, principalmente, cuando tenemos grandes cantidades de datos muy variados entre sí y necesitamos escalabilidad. Esto sería, por ejemplo, el caso de temas complejos de IA como Big Data o el estudio de un gran número de muestras para entrenar una red neural.

En nuestro caso, ni tenemos una gran variedad de datos, ni un gran volumen, ni necesitamos escalabilidad, por lo que sería una pérdida de tiempo realizar el SI con este tipo de BD.

Una base de datos relacional encaja mucho mejor con los requerimientos del SI y aquello que nos piden, por lo que concluimos que **NO** sería adecuado construirlo con una base de datos no relacional, como es MongoDB.

6. Bibliografía

- <https://docs.mongodb.com/>
- <https://pymongo.readthedocs.io/en/stable/>
-