

Proyecto final 2º trimestre



Índice

1.	¿En qué consiste mi proyecto?	2
2.	Estructura de mi base de datos	2
3.	Consultas	3
4.	Mongo Atlas	9
4.1.	Qué es Mongo Atlas	9
4.2.	Cómo se utiliza	9
4.3.	Mongoimport y Mongoexport	16

¿En qué consiste mi proyecto?

He creado una colección “ventas” con la información principal de cada venta de la que he sacado otras tres colecciones con distintos datos cada una,

He utilizado el método aggregate junto con Mongo Atlas y sus herramientas para importar y exportar

Estructura de la base de datos

productos

Storage size: 20.48 kB	Documents: 15	Avg. document size: 196.00 B	Indexes: 1	Total index size: 20.48 kB
----------------------------------	-------------------------	--	----------------------	--------------------------------------

proveedores

Storage size: 20.48 kB	Documents: 4	Avg. document size: 62.00 B	Indexes: 1	Total index size: 20.48 kB
----------------------------------	------------------------	---------------------------------------	----------------------	--------------------------------------

VendedorCliente

Storage size: 20.48 kB	Documents: 10	Avg. document size: 78.00 B	Indexes: 1	Total index size: 20.48 kB
----------------------------------	-------------------------	---------------------------------------	----------------------	--------------------------------------

ventas

Storage size: 20.48 kB	Documents: 10	Avg. document size: 412.00 B	Indexes: 1	Total index size: 20.48 kB
----------------------------------	-------------------------	--	----------------------	--------------------------------------

Consultas

Para crear la colección “productos”

```
db.ventas.aggregate([
  {
    $unwind: {
      path: "$item"
    }
  },
  {
    $project: {
      _id:0,
      id:1,
      nombre:"$item.nombre",
      precioVenta:"$item.precioVenta",
      precioCompra:"$item.precioCompra",
      unidades:"$item.unidades",
      características:"$item.características",
    }
  },
  {
    $out:"productos"
  }
])
```

Para crear la colección “VendedorCliente

```
db.ventas.aggregate([
  {
    $project:{
      _id:0,
      id:1,
      vendedor:1,
      cliente:"$cliente.nombre"
    }
  },
  {
    $out:"VendedorCliente"
  }
])
```

Para crear la colección “proveedores”, la cual está formada por un recuento del número de ventas en las que han proporcionado productos

```
db.ventas.aggregate([
  {
    $project:{
      nombre:{
        $concatArrays:[
          "$item.características.marca"
        ]
      }
    }
  },
  {
    $unwind: {
      path: "$nombre"
    }
  },
  {
    $group:{
      _id: "$nombre",
      nProvisiones: {
        $sum:1
      }
    }
  },
  {
    $project:{
      _id:0,
      nombre:"$_id",
      nProvisiones:"$nProvisiones"
    }
  },
  {
    $sort:{
      provisiones: -1
    }
  },
  {
    $out: "proveedores"
  }
])
```

Para obtener el beneficio generado por cada vendedor ordenado de mayor a menor

```
db.ventas.aggregate([
  {
    $unwind: {
      path: '$item'
    }
  },
  {
    $project: {
      _id: 0,
      vendedor: '$vendedor',
      costeTotal: {
        $multiply: [
          '$item.precioCompra',
          '$item.unidades'
        ]
      },
      dineroTotal: {
        $multiply: [
          '$item.precioVenta',
          '$item.unidades'
        ]
      }
    }
  },
  {
    $group: {
      _id: '$vendedor',
      beneficioTotal: {
        $sum: {
          $subtract: [
            '$dineroTotal',
            '$costeTotal'
          ]
        }
      }
    }
  },
  {
    $sort: {
      beneficioTotal: -1
    }
  }
])
```

Resultado:

```
{ _id: 'Juan', beneficioTotal: 279 },
{ _id: 'Lucas', beneficioTotal: 195 },
{ _id: 'Daniel', beneficioTotal: 128 },
{ _id: 'Manuel', beneficioTotal: 88 }
```

Para obtener la media de dinero generado por cada venta

```
db.ventas.aggregate([
  {
    $unwind: {
      path: '$item'
    }
  },
  {
    $project: {
      _id: 0,
      id: 1,
      dineroTotal: {
        $multiply: [
          '$item.precioVenta',
          '$item.unidades'
        ]
      }
    }
  },
  {
    $group: {
      _id: '$id',
      media: {
        $avg: {
          $sum: '$dineroTotal'
        }
      }
    }
  },
  {
    $project: {
      _id: 1,
      media: {
        $round: ['$media', 0]
      }
    }
  },
  {
    $sort: {
      _id: 1
    }
  }
])
```

Resultado:

```
{_id: 1, media: 220 },
{_id: 2, media: 265 },
{_id: 3, media: 778 },
{_id: 4, media: 309 },
{_id: 5, media: 199 },
{_id: 6, media: 279 },
{_id: 7, media: 394 },
{_id: 8, media: 399 },
{_id: 9, media: 279 },
{_id: 10, media: 220 }
```

Para obtener solo ventas realizadas por Manuel o Lucas que no hayan hecho envío

```
db.ventas.aggregate([
  {
    $match: {
      $and: [{ $or: [{ vendedor: 'Manuel' }, { vendedor: 'Lucas' } ] }, { 'envío': false } ]
    }
  },
  {
    $project: {
      _id: 0,
      id: 1,
      envío: 1,
      vendedor: 1
    }
  }
])
```

Resultado:

```
{ id: 2, 'envío': false, vendedor: 'Manuel' },
{ id: 6, 'envío': false, vendedor: 'Manuel' },
{ id: 7, 'envío': false, vendedor: 'Lucas' }
```


Para obtener las ventas con envío junto con el vendedor y el cliente desde la colección VendedorCliente

```
db.VendedorCliente.aggregate([
  {
    $lookup: {
      from: 'ventas',
      localField: 'id',
      foreignField: 'id',
      as: 'venta'
    }
  }, {
    $unwind: {
      path: '$venta'
    }
  }, {
    $project: {
      _id: 0,
      id: 1,
      vendedor: 1,
      cliente: 1,
      envio: '$venta.envío'
    }
  }, {
    $match: {
      envio: true
    }
  }
])
```

Resultado:

```
{ id: 1, vendedor: 'Lucas', cliente: 'Claudio Quero', envio: true },
{ id: 3, vendedor: 'Daniel', cliente: 'Martin Morcillo', envio: true },
{ id: 4, vendedor: 'Juan', cliente: 'Roberto Patiño', envio: true },
{ id: 9, vendedor: 'Manuel', cliente: 'Alexander Coll', envio: true },
{ id: 10, vendedor: 'Juan', cliente: 'Enrique Pineda', envio: true }
```

Mongo Atlas


¿Qué es MongoDB Atlas?

Es un servicio de bases de datos en la nube que te permite crear y administrar tus bases de datos a través de su plataforma

¿Cómo se utiliza MongoDB Atlas?

El primer paso es crear una cuenta de manera gratuita, para ello puedes utilizar tu cuenta de Google

Empiece gratis
No se necesita tarjeta de crédito

 Regístrese con Google


o

Su empresa (opcional)

Su correo electrónico del trabajo

Nombre

Apellido

Password 

8 caracteres como mínimo

☐ Acepto las [condiciones del servicio](#) y la [política de privacidad](#).

Empiece gratis

¿Ya dispone de una cuenta? [Inicie sesión](#).

Una vez creada tu cuenta, debes crear un usuario para la base de datos, se hace seleccionando el botón “Add new database user” en la pestaña Database Access

Deberás seleccionar el método de autenticación, junto con el nombre de usuario y la contraseña

Authentication Method

Password

Certificate

AWS IAM
(MongoDB 4.4 and up)

MongoDB uses [SCRAM](#) as its default authentication method.

Password Authentication

usuario1

SHOW

Podrás editar también los derechos que quieres otorgar al usuario

Database User Privileges

Configure role based access control by assigning database users a mix of one built-in role, multiple custom roles, and multiple specific privileges. A user will gain access to all actions within the roles assigned to them, not just the actions those roles share in common. **You must choose at least one role or privilege.** [Learn more about roles.](#)

Built-in Role

1 SELECTED



Select one [built-in role](#) for this user.

Read and write to any database



Custom Roles



Select your [pre-defined custom role\(s\)](#). Create a custom role in the [Custom Roles](#) [tab](#).

Specific Privileges



Select multiple privileges and what database and collection they are associated with.
Leaving collection blank will grant this role for all collections in the database.

Restrict Access to Specific Clusters/Data Lakes

Enable to specify the resources this user can access. By default, all resources in this project are accessible.

OFF

Temporary User

This user is temporary and will be deleted after your specified duration of 6 hours, 1 day, or 1 week.

OFF

Desde la pestaña Network Access podrás seleccionar una lista de direcciones IP a las que permitir acceso, en el caso de no querer colocar ningún límite de este tipo se puede seleccionar la opción “Allow access from anywhere” para eliminar esta restricción

×

Add IP Access List Entry

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more.](#)

ADD CURRENT IP ADDRESS

ALLOW ACCESS FROM ANYWHERE

Access List Entry:

Comment:

☐ This entry is temporary and will be deleted in

6 hours ▾


Cancel

Confirm

Tras esto, crearemos la base de datos, para ello iremos a la pestaña Database y seleccionaremos “Build a database”

Seleccionaremos la opción gratuita.

PREVIEW

 **Serverless**


For serverless applications that aren't critical with variable traffic. Minimal configuration required.

- ✓ Pay only for the operations you run
- ✓ Resources scale seamlessly to meet your workload
- ✓ Always-on security and backups

Create

Starting at
\$0.30/1M reads

ADVANCED

 **Dedicated**


For production applications with sophisticated workload requirements. Advanced configuration controls.

- ✓ Network isolation and fine-grained access controls
- ✓ On-demand performance advice
- ✓ Multi-region and multi-cloud options available

Create

Starting at
\$0.08/hr*
*estimated cost \$56.94/month

FREE

 **Shared**

For learning and exploring MongoDB in a cloud environment. Basic configuration options.

- ✓ No credit card required to start
- ✓ Explore with sample datasets
- ✓ Upgrade to dedicated clusters for full functionality


Create


Starting at
FREE


En la siguiente pantalla podremos cambiar distintos ajustes de nuestra base de datos, como la localización y el nombre

Cloud Provider & Region

AWS, Paris (eu-west-3) ▾











★ Recommended region ⓘ 🏷️ Paid tier region ⓘ


NORTH AMERICA

 Oregon (us-west-2) ★


 N. Virginia (us-east-1) ★

 Ohio (us-east-2) ★ 🏷️


 N. California (us-west-1) 🏷️


 Montreal (ca-central-1) 🏷️


SOUTH AMERICA


 Sao Paulo (sa-east-1)


EUROPE


 Paris (eu-west-3) ★

 Ireland (eu-west-1) ★


 Frankfurt (eu-central-1) ★

 Stockholm (eu-north-1) ★


 London (eu-west-2) ★ 🏷️

 Milan (eu-south-1) ★ 🏷️


MIDDLE EAST

 Bahrain (me-south-1) ★


AFRICA


 Cape Town (af-south-1) ★


AUSTRALIA


 Sydney (ap-southeast-2) ★


ASIA


 Seoul (ap-northeast-2)

 Singapore (ap-southeast-1) ★

 Hong Kong (ap-east-1) ★

 Mumbai (ap-south-1)

 Tokyo (ap-northeast-1) ★

 Osaka (ap-northeast-3) ★ 🏷️

Cluster Tier

M0 Sandbox (Shared RAM, 512 MB Storage) ^{Encrypted} ▴

Additional Settings

MongoDB 5.0, No Backup ▴

Cluster Name

Cluster0 ▴

FREE

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

[Back](#)

Create Cluster

Ahora debemos seleccionar el tipo de autenticación, junto con el método y lista de acceso, una vez hayamos seleccionado el usuario y la lista creados anteriormente pinchamos en el botón “Finish and close” para crear la base de datos

✓ 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

Username	Authentication Type	
usuario1	Password	 EDIT

Ahora podremos conectar nuestra base de datos local con la base en la nube que acabamos de crear, para ello, iremos a la sección “Connect” en nuestra base de datos

Cluster0

Connect

View Monitoring

Browse Collections

...

Seleccionaremos "Connect using MongoDB Compass"

Connect to Cluster0


✓ Setup connection security


Choose a connection method


Connect

Choose a connection method [View documentation](#)

Get your pre-formatted connection string by selecting your tool below.

**Connect with the MongoDB Shell**
Interact with your cluster using MongoDB's interactive Javascript interface

**Connect your application**
Connect your application to your cluster using MongoDB's native drivers

**Connect using MongoDB Compass**
Explore, modify, and visualize your data with MongoDB's GUI

Copiaremos el enlace que nos proporciona y lo usaremos como enlace de conexión en MongoDB Compass, reemplazando <password> por la contraseña de nuestro usuario

✓ Setup connection security

✓ Choose a connection method

Connect

I do not have MongoDB Compass

I have MongoDB Compass

1 Choose your version of Compass:

1.12 or later

See your Compass version in "About Compass"

2 Copy the connection string, then open MongoDB Compass.

mongodb+srv://usuario1:<password>@cluster0.x2qcy.mongodb.net/test

You will be prompted for the password for the **usuario1** 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](#)

New Connection

☆ FAVORITE

Fill in connection fields individually

Paste your connection string (SRV or Standard ⓘ)

mongodb+srv://usuario1:<password>@cluster0.x2qcy.mongodb.net/test

Connect

En el caso de querer conectarnos desde MongoDB Shell seleccionaremos “Connect with the MongoDB Shell” y utilizamos el enlace que nos proporciona, tras introducir el enlace nos pedirá escribir la contraseña, una vez lo hagamos tendremos acceso a la base de datos

2 Run your connection string in your command line

Use this connection string in your application:

```
mongosh "mongodb+srv://cluster0.x2qcy.mongodb.net/myFirstDatabase" --apiVersion 1 --  
username usuario1
```

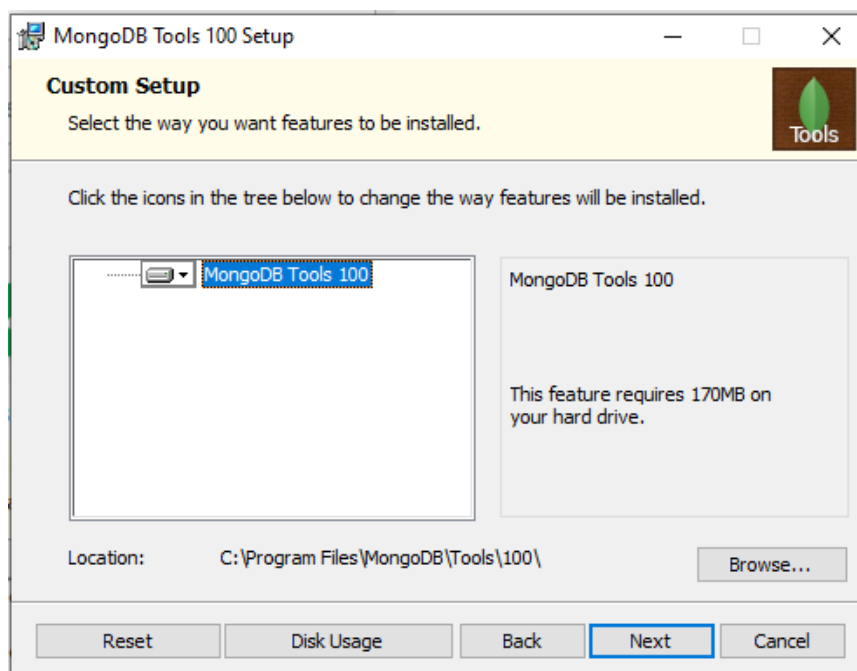
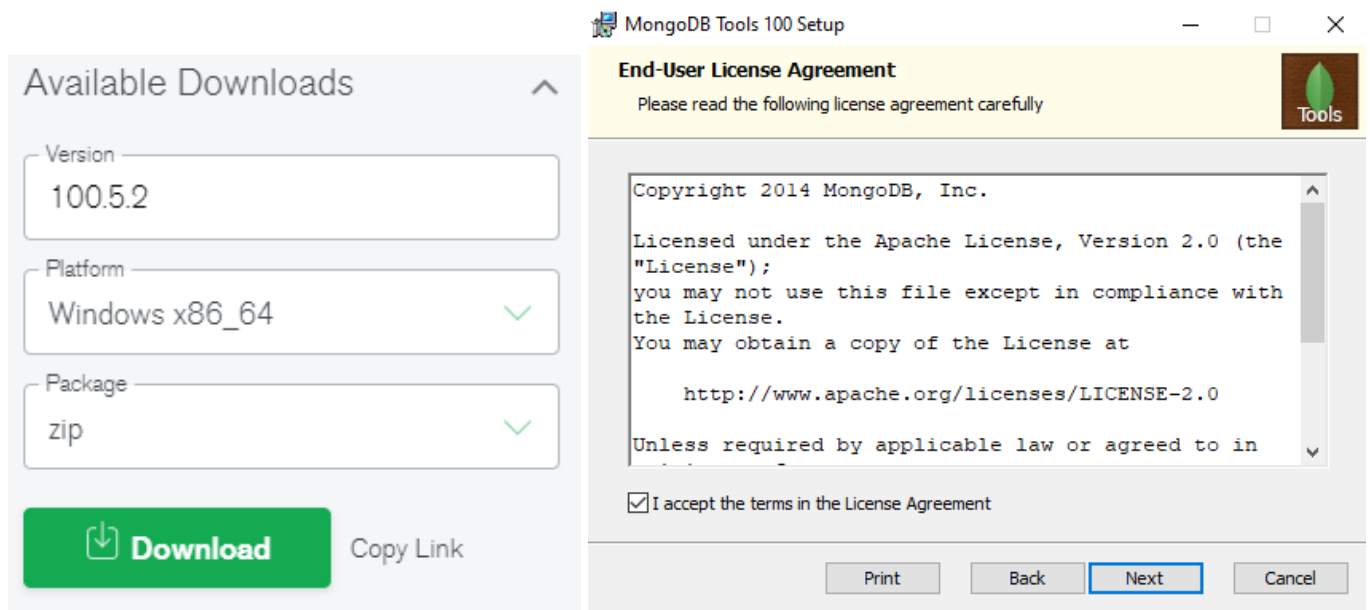


Mongoimport y Mongoexport

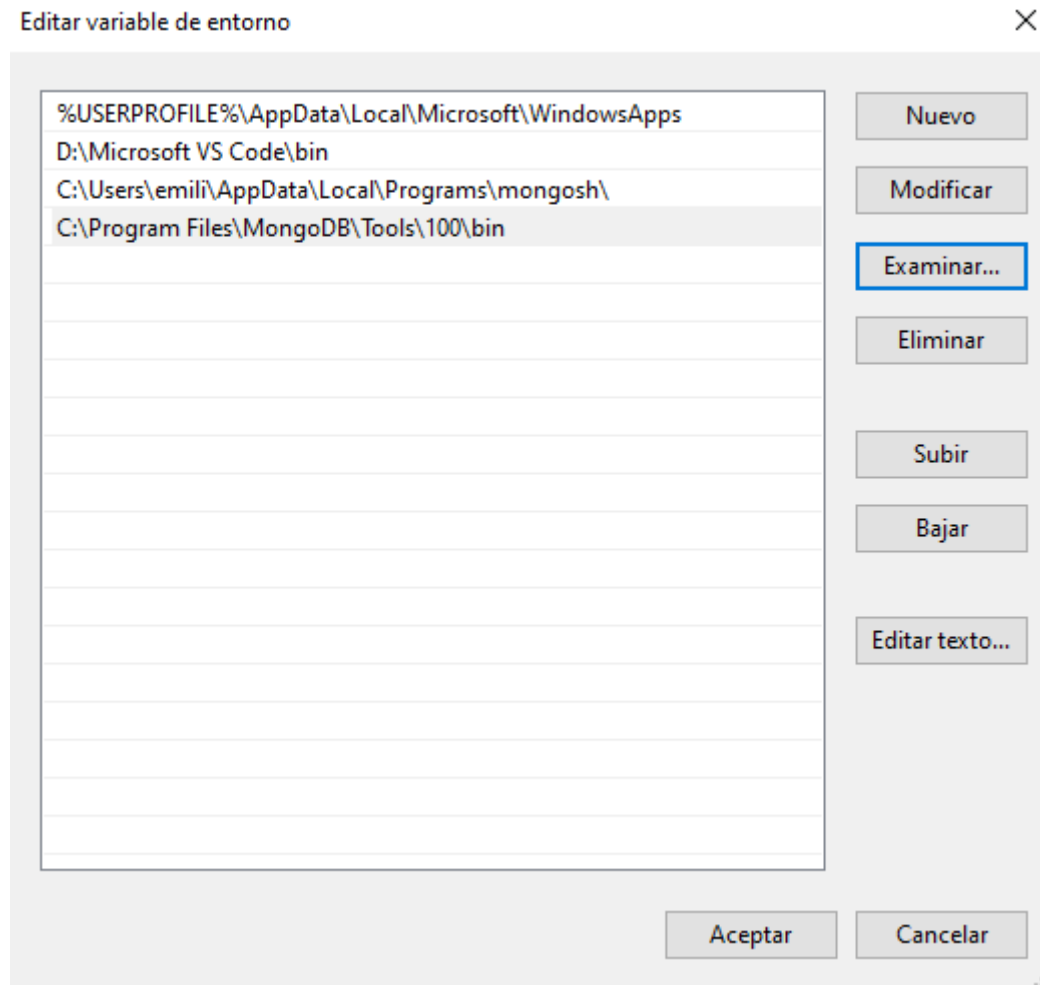
Primero debemos instalar las “Database Tools”, las cuales nos permitirán usar mongoimport y mongoexport para poder mover datos entre varias bases de datos

Instalación de Database Tools

Debemos descargar el archivo msi, aceptar la licencia y finalmente elegir la ruta en la que queremos que se descarguen las herramientas



Tras la instalación debemos añadirlo a las variables de entorno del sistema




Exportar datos

Desde la línea de comandos:

Como local

Sintaxis:

mongoexport --collection=<colección> --db=<base de datos> --out=<nombre de archivo>.json

 productos

07/03/2022 6:29

Archivo de origen ...

4 KB

Desde MongoDB Atlas

Sintaxis:

```
mongoexport --uri  
mongodb+srv://usuario1:<password>@cluster0.x2qcy.mongodb.net/test --collection  
<colección> --type json --out <nombre de archivo>.json
```


Importar datos

Desde la línea de comandos:

Como local

Sintaxis:

```
mongoimport --db=<base de datos> --collection=<colección> --out=<nombre de archivo>.json
```

 productos

07/03/2022 6:29

Archivo de origen ...

4 KB

Desde MongoDB Atlas

Sintaxis:

```
mongoimport --uri  
mongodb+srv://usuario1:<password>@cluster0.x2qcy.mongodb.net/test --collection  
<colección> --type json --file <nombre de archivo>.json
```