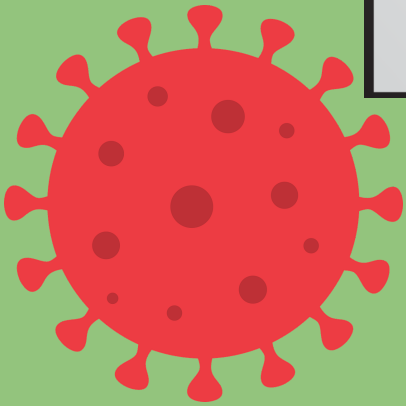


PROYECTO SEGUNDA EVALUACIÓN

COVID_19
Base_de_Datos

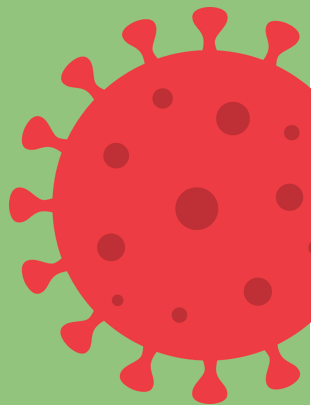


Nombre: José Miguel

Apellidos: Rodríguez Muñoz

Fecha: 06/03/2022

Módulo: Base de datos





Índice



Operadores nuevos y usados



Consultas aggregate



Explicación Mongo Atlas



Explicación Mongoexport e import

Operadores nuevos



\$cmp – Compara dos valores y devuelve:

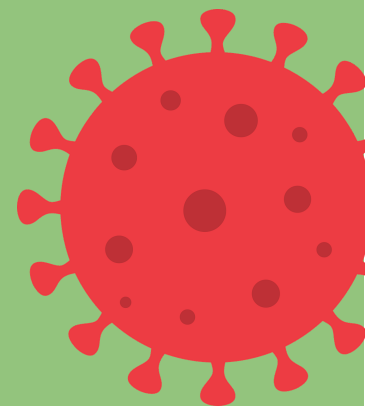
- -1 si el primer valor es menor que el segundo.
- 1 si el primer valor es mayor que el segundo.
- 0 si los dos valores son equivalentes.

\$cond – Evalúa una expresión booleana para devolver una de las dos expresiones de devolución especificadas.

Si se <boolean-expression>evalúa como true, entonces \$condevalúa y devuelve el valor de la <true-case>expresión. De lo contrario, \$condevalúa y devuelve el valor de la <false-case>expresión.

Operadores usado

- \$unwind
- \$subtract
- \$sort
- \$round
- \$match
- \$divide
- \$multiply
- \$addField
- \$avg
- \$lookup
- \$arrayElemAt
- \$out



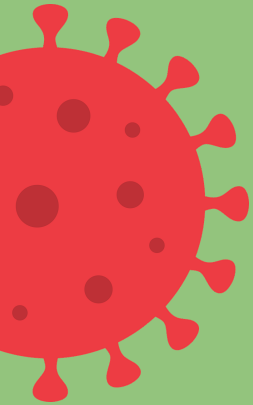
CONSULTAS AGGREGATE

La realización de esta consulta tiene el fin de dar por comunidades el descuento producido en la vacuna pfizer.

En el inicio de la consulta utilizamos \$unwind para que en esta etapa se desconstruya el campo matriz "vacuna". Usando \$project pasa los documentos con los campos solicitados a la siguiente etapa de la canalización, en este caso "vacunas", "comunidad" e "id" dándoles valor 1 y 0 se lo damos a la "_id" para que no aparezca. Luego, utilizando \$cond obtenemos las dos expresiones diferentes dado que devuelve el valor dependiendo de la condición especificada en ella donde tiene un operador \$gte que pide que sea mayor o igual a "19", si esto se cumple devolverá 30 y sino 20. Más adelante tenemos \$match que filtra los documentos con el valor especificado en este caso que tenga el valor "Pfizer". Y en la última etapa tenemos un \$project que pasa los campos "descuentos", "comunidad" e "id", que el campo "name", excluya al campo "vacunas".

```
db.vacunas.aggregate( [
  { $unwind:{path:"$vacunas"} },
  { $project: { _id:0, vacunas: 1, id:1, comunidad:1, descuento: {
    $cond:{if:{$gte:["$vacunas.coste",19]},then:30,else:20} } } },
  { $match : { "vacunas.name": "Pfizer" } },
  { $project: { id:1, name:"$vacunas.name", descuento:1,
    comunidad:1 } }
]).pretty()
```





El fin de esta consulta es saber qué vacunas son efectivas
Al principio de la consulta utilizamos \$unwind para que
desconstruya el "campo vacunas".

A continuación, usamos \$project para que sean visibles
los campos "vacunas" e "id", también usamos el operador
\$cmp para que uno devuelva uno de los tres valores
(-1,1,0), obteniendo así la información de si son eficaces
las vacunas.

Luego, tenemos el operador \$project para que el campo
"name" se libere del campo "vacunas", y el campo
"eficacia_porcentaje" se libere de "vacunas.caracteristicas",
dando el "name", la "eficacia_porcentaje", la "efectividad"
y la "id".

```
db.vacunas.aggregate( [  
  { $unwind:"$vacunas" },  
  { $project: { id:1, _id: 0, vacunas: 1,  
    efectividad:{$cmp:["$vacunas.caracteristicas.eficacia_porcenta  
e", 90] } } },  
  {$project:{name:"$vacunas.name",id:1,eficacia_porcentaje:"$va  
cunas.caracteristicas.eficacia_porcentaje",efectividad:1}}  
]).pretty()
```



El objetivo de esta consulta es dar un documento con los días de importe de las vacunas que sean superior o igual al 2021-12-00 y menor o igual al 2022-03-00, con el coste total de las vacunas y sus medias.

Primero, usamos \$unwind para deconstruir el campo matriz "vacunas".

A continuación, se usa \$match para filtrar las fechas que sean que sean superior o igual al "2021-12-00" y menor o igual al "2022-03-00" con los operadores \$gte y \$lte.

Luego, utilizamos \$group para agrupar los documentos por fecha y calcula el total del coste de la vacuna, además de la cantidad promedio del coste, usando los operadores \$sum, \$multiply y \$avg para sumar y multiplicar el "coste" con las "dosis recibidas", además con este se calcula las medias recibidas. Por último, tenemos el operador \$sort ordena los resultados por id de la fecha para cada grupo en orden ascendente.

```
db.vacunas.aggregate( [
  { $unwind:"$vacunas" },
  {$match:{ "vacunas.Dia_de_importe": { $gte: new ISODate("2021-12-00"),
    $lte: new ISODate("2022-03-00") } } },
  {$group:{_id:{$dateToString:{format:"%Y-%m-%d",date:"$vacunas.Dia_de_i
    mporte" } }},
  totaldecoste:{$sum:{$multiply:["$vacunas.coste","$vacunas.dosis_recibidas"
    ] } }, cantidadmedia: { $avg: "$vacunas.dosis_recibidas" } } },
  { $sort : { _id:1 } }
] ).pretty()
```

Con esta consulta se pretende crear una ficha de transporte en la que aparezca el transportador, el coste, el tiempo medio de llegada, la primera envío de vacunas y el último, y saber que descuento se le aplica al coste .

Se usa `$project`, para que sea visible los campos "transportador", "costetrans", además ocultamos la "_id", también tenemos el operador `$arrayElemAt` que devuelve el elemento en el índice de matriz especificado, de esta manera devuelve el primer y último elemento de la "envíos" matriz.

Posteriormente, usamos otro `$project` para dejar visible los campos "transportador", "first", "last", "tiempodellegada" y "costetrans", dejando la "_id" no visible, aparte tenemos el operador `$cond` que tiene un operador `$gte` que pide que sea mayor o igual a "15", si esto se cumple devolverá 40 y sino 10. Tenemos otro `$project`, dejando los mismos campos visibles, aportando también el campo "descuentos" para que sea visible, aunque hay otra operación en la que se calcula el descuento del transporte, usando el operador `$round` para redondear el valor, el `$divide` para dividir entre 100 el valor resultante de la multiplicación del descuento y el coste con `$multiply`. Por último tenemos `$out` para que nos dirija el documento a la base "test y de colección "docxxtransporte".

```
db.vacunas.aggregate( [
  { $project: { transportador:1, costetrans:1, _id:0,
    first: { $arrayElemAt: [ "$envios", 0 ] },
    last: { $arrayElemAt: [ "$envios", -1 ] },
    tiempodellegada: { $avg: "$Tiempollegada", } },
  { $project: { transportador:1, first:1, last:1, tiempodellegada:1, costetrans:1, _id:0,
    descuentos: { $cond: { if: { $gte: [ "$tiempodellegada", 15 ] }, then: 40, else: 10 } } },
  { $project: { transportador:1, first:1, last:1, tiempodellegada:1, descuentos:1, costetrans:1, _id:0,
    descuentodeltrans: { $round: [ { $divide: [ { $multiply: [ "$descuentos", "$costetrans" ] }, 100 ] }, 0 ] } },
  { $out: { db: "test", coll: "docxxtransporte" } }
]).pretty()
```



En esta consulta, queremos actualizar una base de datos por comunidad en la que se incluya si las dosis recibidas han sido desperdiciadas, los informes de la actividad de la vacuna y los de transporte.

Primero usamos \$unwind para deconstruir el campo matriz "vacunas".

Después usamos \$project para dejar visible los campos "transportador", "id", "vacunas", "comunidad" y no visible la "_id", hay un operador \$round para redondear el resultado, de la multiplicación con \$multiply de las "dosis recibidas" por el "coste" y las "dosis administradas" por el "coste", restandose al final con \$subtract.

Luego usamos otro \$project, que deja los mismos campos visibles y excluidos, además hay un operador \$cmp que determina si las vacunas son desperdiciadas o no, si el resultado es menor que 0 se data que las dosis han sido desperdiciadas (=1) sino es que no habrá sobrado (=1) o se han acabado (=0).

A continuación, hay otro \$project que dejará visible los campos "id", "transportador", "vacunas desperdiciadas", "comunidad" y el "name" se liberará de la array de documentos "vacunas" al que pertenece.

Más adelante se utiliza dos veces el operador \$lookup la primera vez para unir los datos sacados con la colección "información de actividad", convirtiéndose esta en una array de documentos, nombrada con la función "as", pero habrá que decirle cuál es su conexión de campos en el "localField" (colección de aggregate) y "foreignField" (colección que se quiere implementar), el otro \$lookup unirá lo creado antes con la colección creada anteriormente, unidas por el campo "transportador". Más adelante con \$addFields y \$arrayElemAt, hacemos que el campo con array que había salido se le quite los "[]" y aparezca con "{}", y borrarémos el campo con \$unset para renombrarlo.

Y por último habrá un \$out que creará una nueva base de datos "documento" y una colección "informe".

```
db.vacunas.aggregate( [ { $unwind:"$vacunas" }, { $project: { comunidad:1, id:1, _id: 0, vacunas: 1, transportador:1, dosisdesperdiciadas:{
```



```
$round:[{$subtract:[{$multiply:["$vacunas.dosis_recibidas","$vacunas.coste"
]], {$multiply:["$vacunas.dosis_administradas","$vacunas.coste"]}}] } }, {
$project: { comunidad:1, id:1, _id: 0, vacunas: 1, transportador:1,
vacunasdesperdiciadas: { $cmp: [ "$dosisdesperdiciadas", 0] } } }, {
$project: { name:"$vacunas.name", comunidad:1, vacunasdesperdiciadas:1,
id:1, transportador:1, } }, { $lookup: { from:
"Informedeactividaddelprocesodevacunación", localField: "id", foreignField:
"_id", as: "totaldatosdevacunaporcomunidad" } }, { $lookup: {
from:"docxxtransporte", localField:"transportador",
foreignField:"transportador", as:"datostransporte" } }, { $addFields:{
datostransportes:{ $arrayElemAt:["$datostransporte",0] } } }, {
$unset:"datostransporte" }, { $out:{db:"documento",coll:"informe"} } ]
).pretty()
```

-Con este comando se puede ver la colección creada sin necesidad de utilizar **Mongocompass**

```
db.getSiblingDB("documento").informe.find().pretty()
```

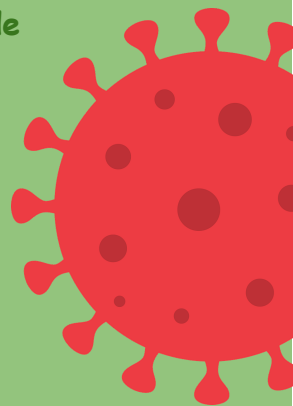


El fin de esta consulta ,es obtener cuales comunidades están inmunizadas,con la intención de ver qué vacunas tienen una duración larga,para poder reforzarlas con un nuevo componente.

Usamos \$unwind para deconstruir el campo matriz "vacunas".

Luego usamos \$project para hacer visible los campos "comunidad", "name", y "composicion", ocultando la "_id",aparte hay \$cond para saber si la vacuna es duradera o no,siendo la condición que tiene que ser mayor o igual a 180 días para que sea duradera.El .pretty() se usa para que devuelva los datos en un formato que sea más fácil de leer.Después,tenemos el operador \$addFields para reemplazar el valor de una array por otro.En la siguiente tenemos otro \$project para dejar visible los campos "comunidad", "tiempodeduracion", "name" y liberar del array de documentos al campo "composicion".Y por último, un \$project que dejara visible los campos de la anterior etapa,pero cambia el nombre del campo "composicion" por el de "nuevocomponente".

```
db.vacunas.aggregate( [  
  { $unwind:"$vacunas" },  
  { $match:{"población_inmunizada":true} },  
  { $project:{_id:0,comunidad:1,"vacunas.name":1,  
    "vacunas.caracteristicas.composicion":1,tiempodeduracion:{  
      $cond:{if:{$gte:["$vacunas.caracteristicas.dias_de_inmunidad_aprox",180]},  
      then:"Si es duradera",else:"No es duradera"} } } },  
  { $addFields: {"vacunas.caracteristicas.composicion": "ALC" } },  
  { $project: { _id:0, comunidad:1, "vacunas.name":1, tiempodeduracion:1,  
    composicion:"$vacunas.caracteristicas.composicion" } },  
  { $project: { _id:0, comunidad:1, "vacunas.name":1, tiempodeduracion:1,  
    "nuevocomponente":"$composicion" } }  
])pretty()
```



Connect to MongoAtlas-Compass

1 Le damos a connec to,y elegimos con que queremos conectarlo,en este caso con 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

>

Go Back

Close

2 A continuación, ponemos que tenemos MongoDB Compass instalado y copiamos el código.

×

Connect to Cluster0

✓ 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.a4ql5.mongodb.net/test

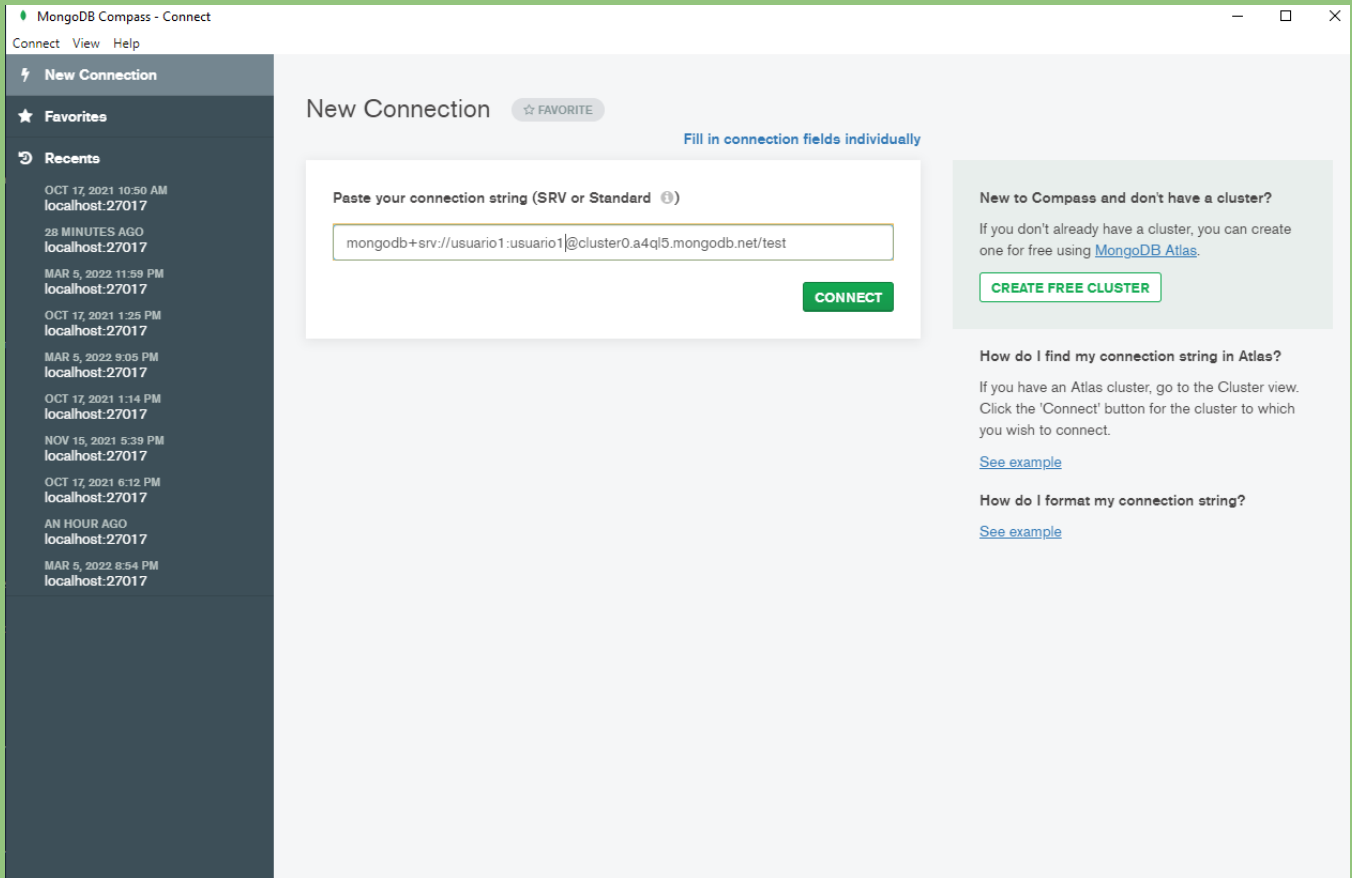
You will be prompted for the password for the **usuario1** user's (Database User) username.
When entering your password, make sure that any special characters are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

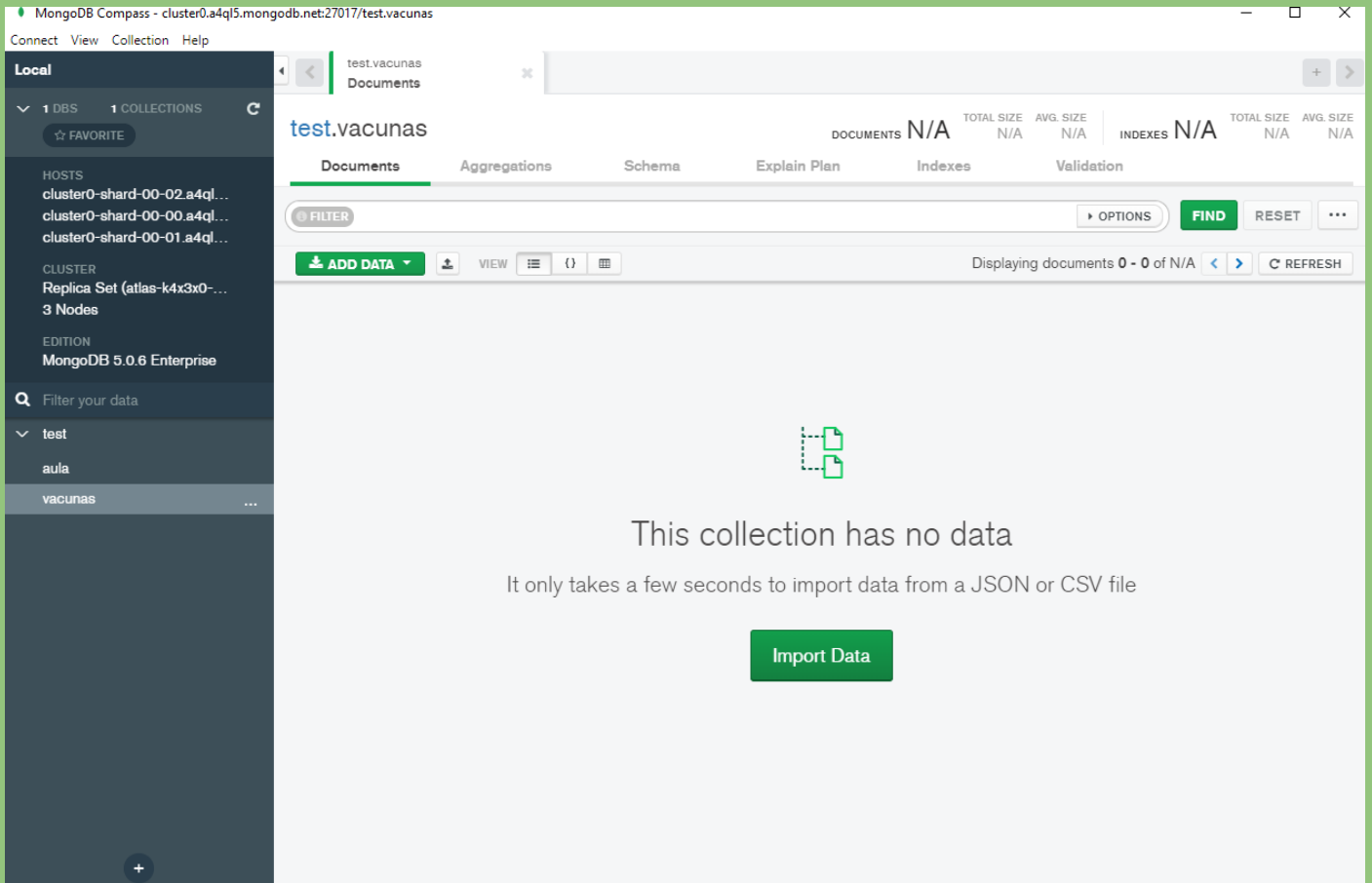
Go Back

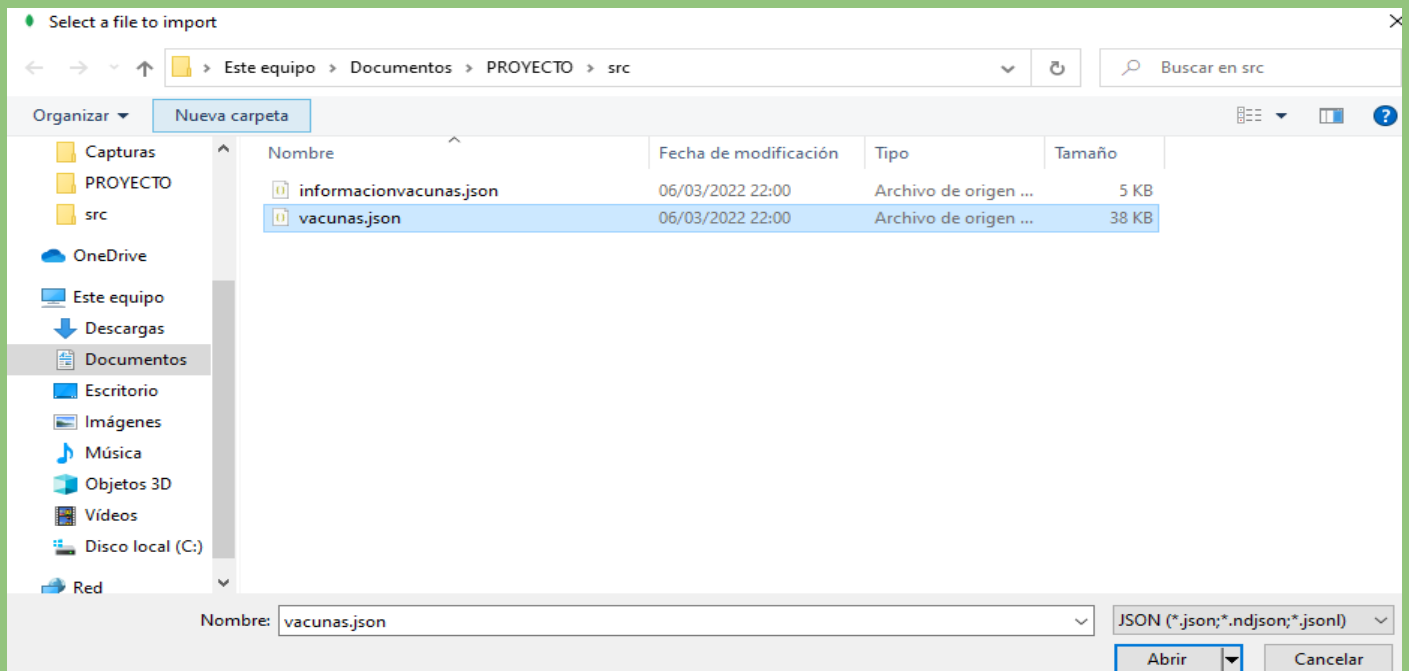
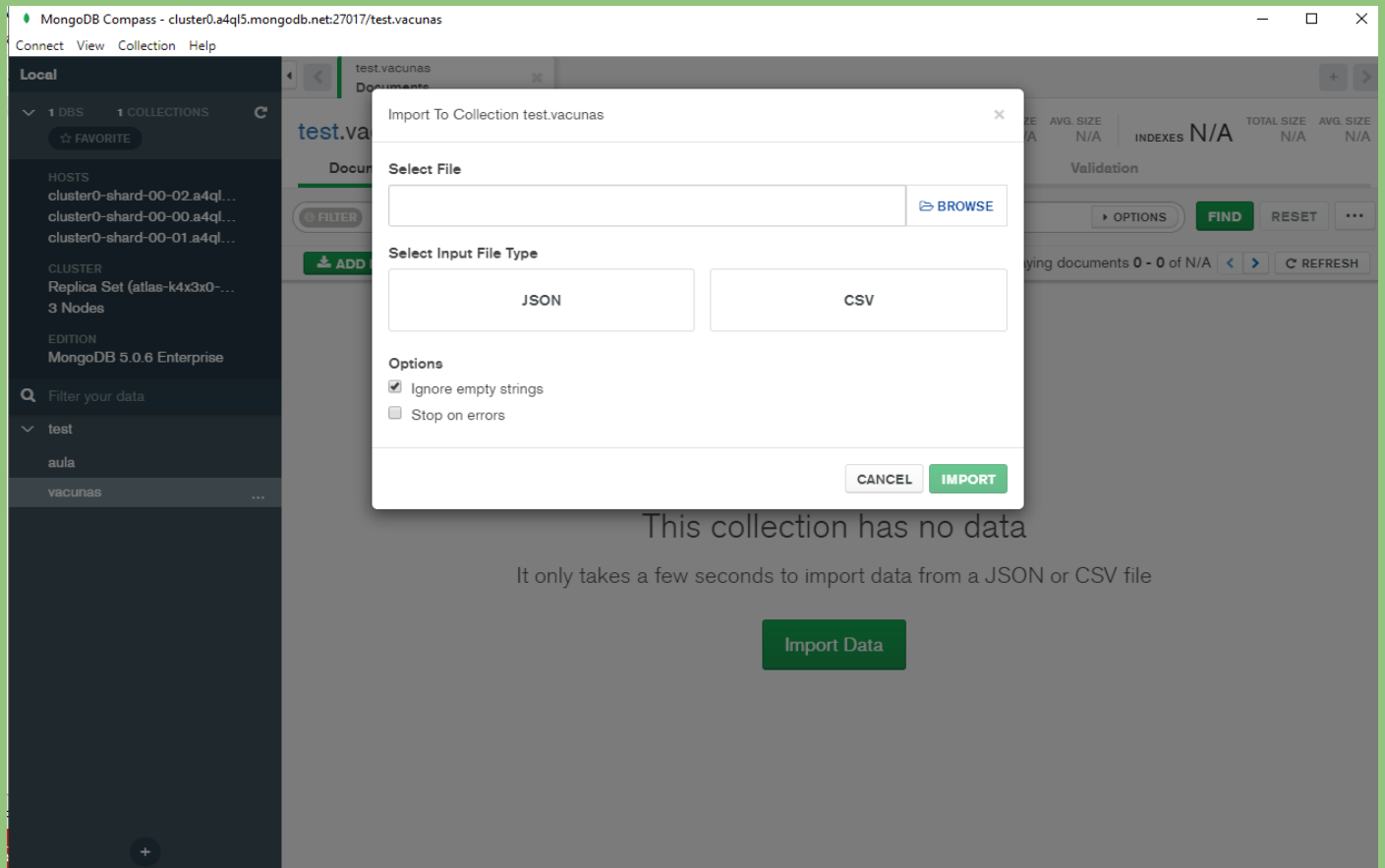
Close

3 Colocamos en código en MongoCompass y modificamos en password poniendo la contraseña creada con el usuario1.



4 Creamos una colección e importamos los datos json, exportados en esa colección creada, donde pone "Import data".





Import To Collection test.vacunas



Select File

C:\Users\Miguel\Documents\PROYECTO\src\vacunas.json

BROWSE

Select Input File Type

JSON

CSV

Options

☒ Ignore empty strings

☐ Stop on errors

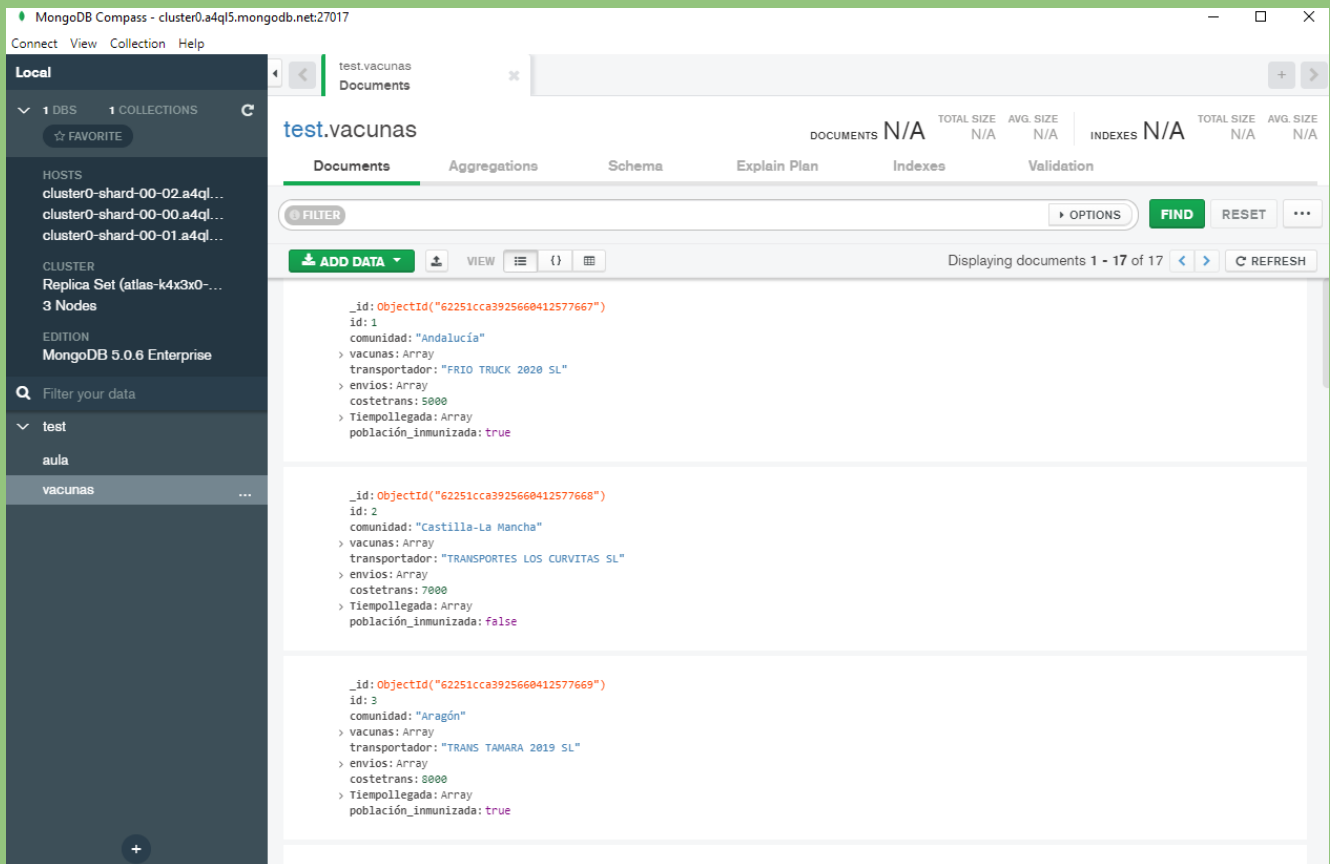
Specify Fields and Types

	<input checked="" type="checkbox"/> <u>i</u> d ObjectID ▼	<input checked="" type="checkbox"/> id Int32 ▼	<input checked="" type="checkbox"/> comunidad String ▼	<input checked="" type="checkbox"/> vacunas String ▼
1	62251cca3925660412577667	[object Object]	Andalucía	[{"name":"Pfizer","dosis
2	62251cca3925660412577668	[object Object]	Castilla-La Mancha	[{"name":"AstraZeneca","
3	62251cca3925660412577669	[object Object]	Aragón	[{"name":"Moderna","dosi
4	62251cca392566041257766a	[object Object]	Asturias	[{"name":"Pfizer","dosis
5	62251cca392566041257766b	[object Object]	Baleares	[{"name":"Pfizer","dosis
6	62251cca392566041257766c	[object Object]	Canarias	[{"name":"Pfizer","dosis
7	62251cca392566041257766d	[object Object]	Cantrabia	[{"name":"Astrazeneca","
8	62251cca392566041257766e	[object Object]	Castilla y León	[{"name":"Pfizer","dosis
9	62251cca392566041257766f	[object Object]	Cataluña	[{"name":"Pfizer","dosis

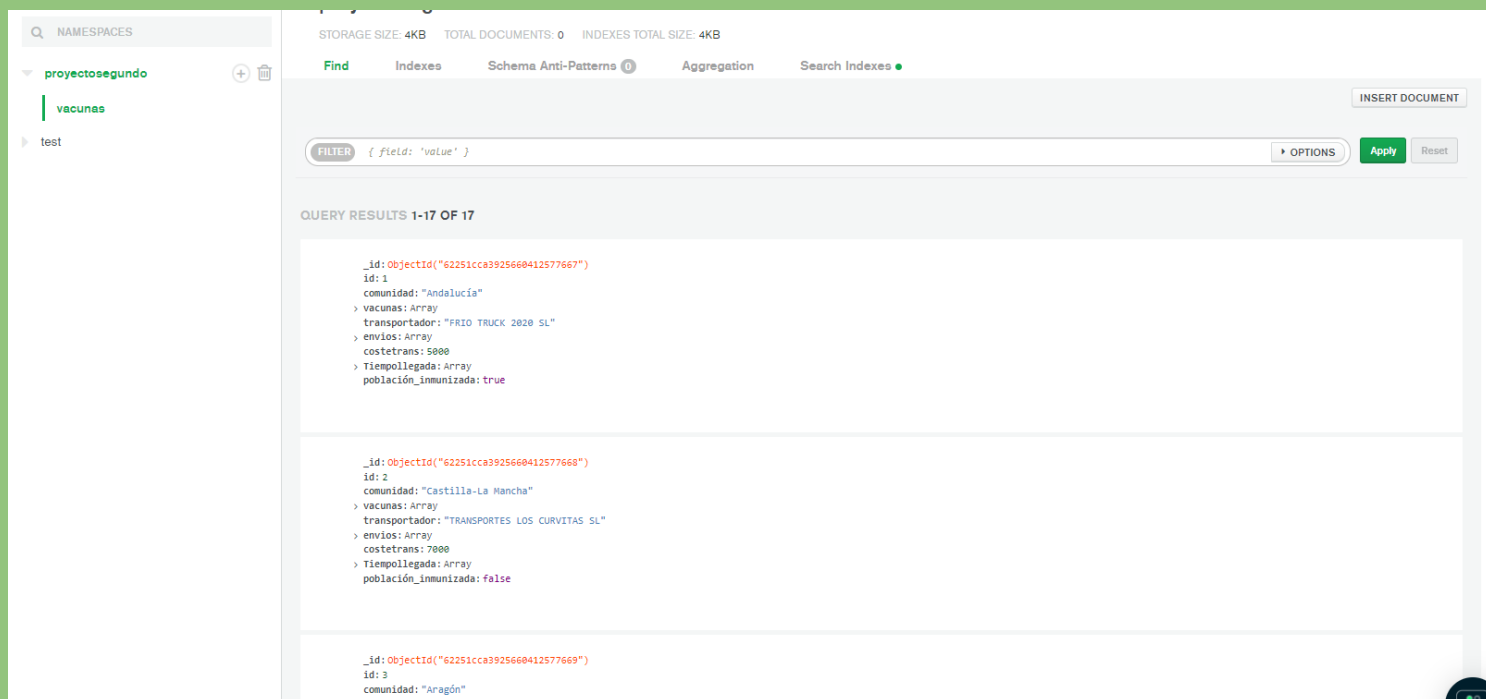
CANCEL

IMPORT

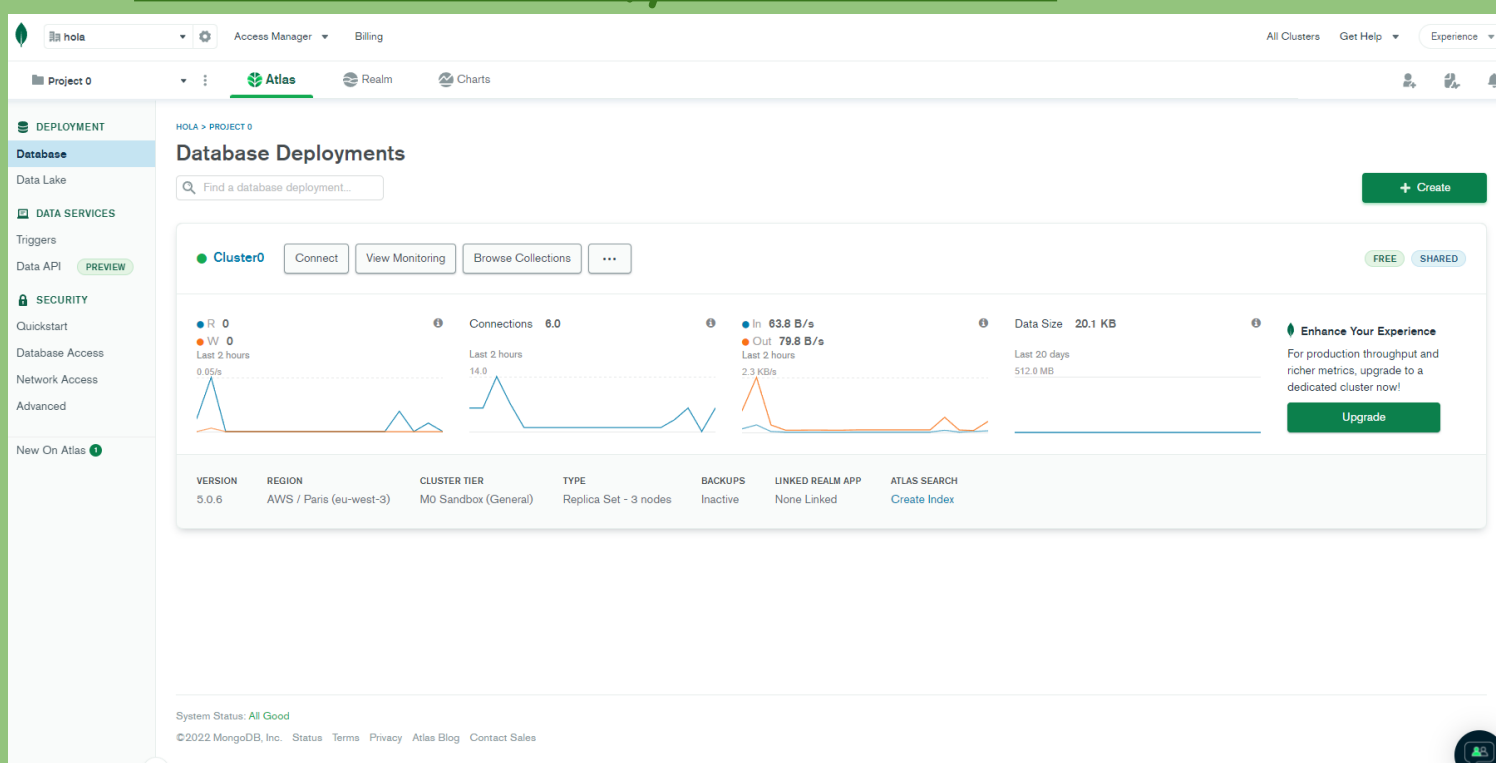
Luego le damos a "import", igual con la otra colección.



A continuación, nos vamos a MongoAtlas y creamos una nueva base de datos llamada "proyectosegundo", aparte de una colección llamada igual que en la de MongoCompas. Como ya tenemos los datos subido desde Compass lo copiamos.



Y así igual con la otra colección, solo dándole a "Browse collections", y a create base.



×

Create Database

Database name ?

proyectosegundo

Collection name ?

PROYECTO|

Additional Preferences

☐ Capped Collection i

☐ Time Series Collection i

Cancel

Create

En "insert to" pondremos los documentos de nuestra colección en json.

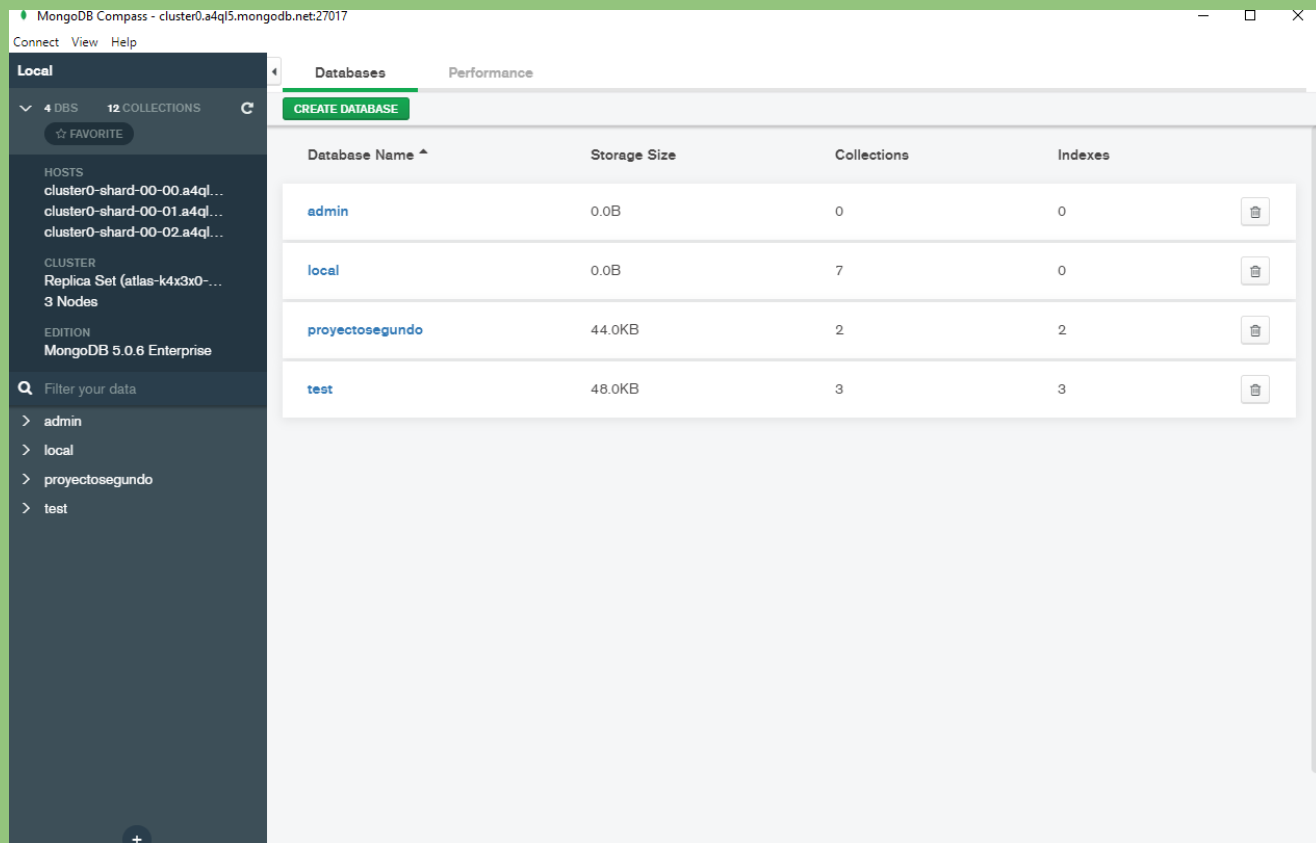
Insert to Collection

VIEW {} ≡

```
1  ▾ [{
2    "_id": 1,
3    "Población_vacunar_Mayor_de_doce_años": 7451975,
4    "Nº_Personas_con_al_menos_1_dosis": 6906661,
5    "Nº_Personas_con_pauta_completa": 6795375,
6    ▾ "Fecha_de_la_última_vacuna_registrada": {
7      "$date": "2021-12-02T00:00:00Z"
8    },
9    "porcentaje_de_efectividad": 70
10 ▾ }, {
11   "_id": 2,
12   "Población_vacunar_Mayor_de_doce_años": 1808784,
13   "Nº_Personas_con_al_menos_1_dosis": 1632952,
14   "Nº_Personas_con_pauta_completa": 1570780,
15   ▾ "Fecha_de_la_última_vacuna_registrada": {
16     "$date": "2021-12-01T00:00:00Z"
17   },
18   "porcentaje_de_efectividad": 60
19 ▾ }, {
20   "_id": 3,
21   "Población_vacunar_Mayor_de_doce_años": 1185134,
22   "Nº_Personas_con_al_menos_1_dosis": 1075919,
```

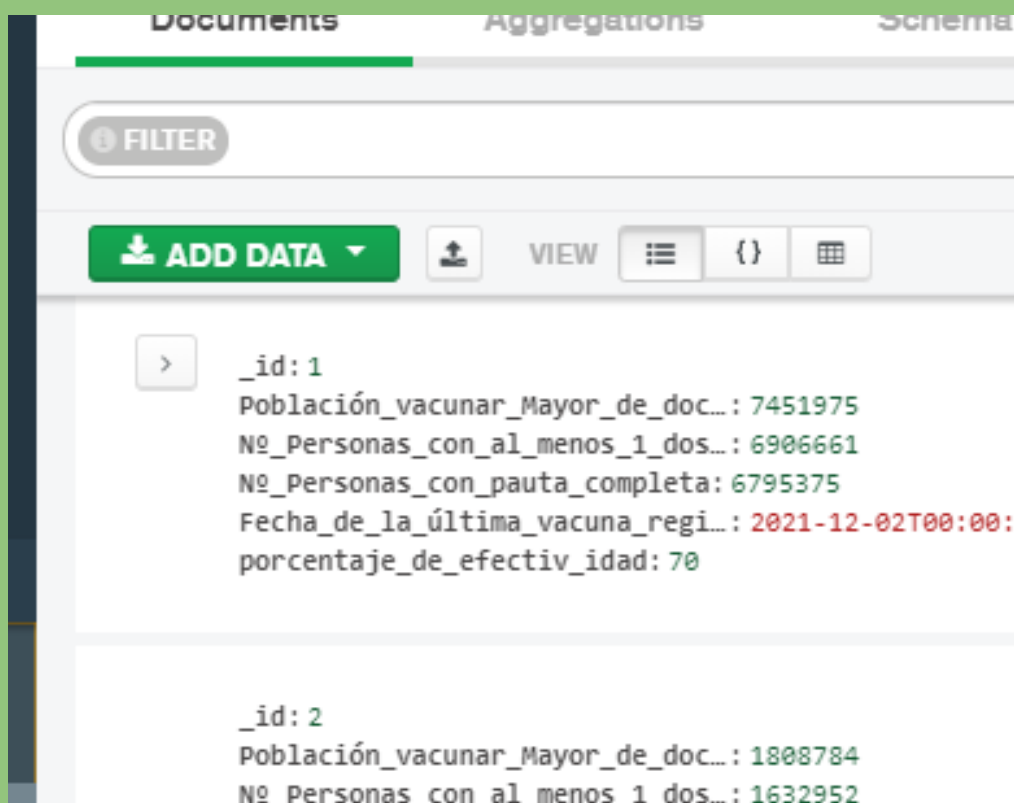
Cancel

Insert



Mongoexport con MongoCompass

Le damos a la flecha hacia arriba y exportamos la colección completa, dando en formato csv.



Export Collection test.vacunas

Export query with filters — 17 results (Recommended)

```
db.vacunas.find(
  {}
)
```

Export Full Collection

CANCEL

SELECT FIELDS

Export Collection test.vacunas

Select Export File Type

JSON

CSV

Output

C:\Users\Miguel\Documents\PROYECTO\src\vacunas.csv

BROWSE

< BACK

CANCEL

EXPORT

Conexión con shell

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> mongo "mongodb+srv://cluster0.a4q15.mongodb.net/proyectoSegundo" --username usuario1
MongoDB shell version v5.0.3
Enter password:
connecting to: mongodb://cluster0-shard-00-02.a4q15.mongodb.net:27017,cluster0-shard-00-00.a4q15.mongodb.net:27017,cluster0-shard-00-01.a4q15.mongodb.net:27017/proyectoSegundo?authSource=admin&compressors=disabled&gssapiServiceName=mongodb&replicaSet=atlas-k4x3x0-shard-0&ssl=true
Implicit session: session { "id" : UUID("a9447c02-6e44-4f8e-8eb1-7d21e4311d3b") }
MongoDB server version: 5.0.6
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
WARNING: some history file lines were truncated at 4095 bytes.
MongoDB Enterprise atlas-k4x3x0-shard-0:PRIMARY>
```

Mongoexport Atlas con shell

```
mongoexport --uri  
mongodb+srv://usuario1:usuario1@cluster0.a4ql5.mongodb.net/proyectosegundo  
--collection vacunas --type CSV  
--outC:\Users\Miguel\Documents\PROYECTO\src\data\vacunas.json
```

Mongoimport Atlas con shell

```
mongoimport --db=proyectosegundo --collection=vacunas --type=CSV  
--file=C:\Users\Miguel\Documents\PROYECTO\src\Export\vacunas  
.json
```

Mongoexport local

```
mongoexport --collection=vacunas --db=proyectosegundo  
--out=C:\Users\Miguel\Documents\PROYECTO\src\Export\project  
osegundo.json
```

Mongoimport local

```
mongoimport --db=proyectosegundo --type=csv --headerline  
--file=C:\Users\Miguel\Documents\PROYECTO\src\Export\project  
osegundo.csv
```

