PROYECTO FINAL 2º TRIMESTRE

Mi proyecto trata de una base de datos de un centro de estética real llamado Mariquilla el cual pertenece a mi amiga a la que le he hecho una prueba de como seria la base de datos de su pequeño negocio .

En este proyecto mostrare los aspectos teóricos y técnicos que he realizado

Aspectos teóricos de los nuevos operadores que he usado.

Método **agregate**:(es un array de etapas)

1. Consiste en una o mas etapas que procesan los documentos.
2. La entrada de una etapa es la salida de la etapa anterior.
3. La primera etapa suele ser un filtro.(todos los documentos y campos de la coleccion)
4. La segunda etapa tiene que superar el filtro y pasaria a la siguiente etapa.
5. Cada etapa se caracteriza o se define por un caraterizador de etapas.

$NOMBRE= el dolar agrupa el campo indicado

$group= id esta reservado a group(es obligatorio)

```
{
  $group:
    {
      _id: <expression>, // Group By Expression
      <field1>: { <accumulator1> : <expression1> },
      ...
    }
 }
```

<expression> primer principio por el que agrupamos
*** <field1> este campo no existe***

ACUMULADOR: OPERACIÓN QUE SE HACE SOBRE UN GRUPO DE VALORES, SEGÚN EL CRITERIO ESTABLECIDO.
$SUM CON VALOR 1= UNO POR CADA UNO DE LOS DOCUMENTOS QUE SE AGRUPA.

$avg
$min
$max
$sum

Otra forma de usar $gt:[v1, v2] que valor 1 sea mayor que valor 2.
$year: operador que actua sobre fecha
$round [] redondea a tantas cifras decimales.

$lookup
Los campos de la etapa anterior son los que le entran a $lookup

```
{
  $lookup:
    {
      from: <colección para unirse>,
      localField: <campo de los documentos de entrada>,
```

```
      foreignField: <campo de los documentos de la colección "from">,
      as: <campo de matriz de salida>
    }
}
```

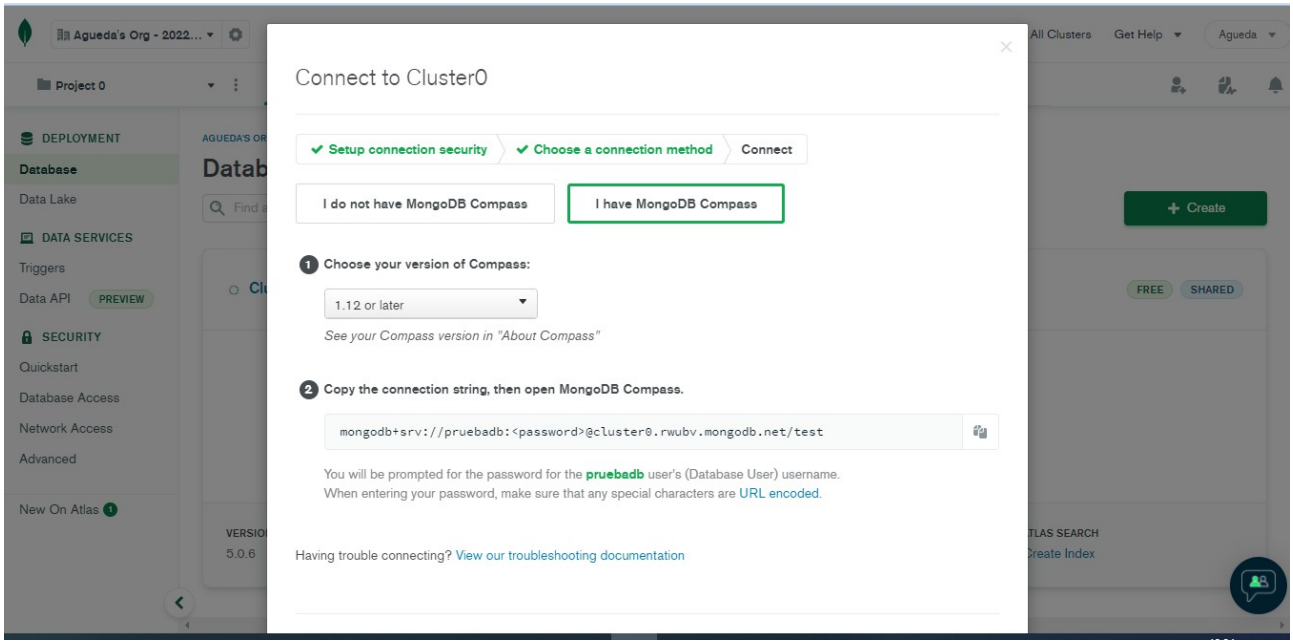
```
{
 from: 'autoventas',
 localField: 'id',
 foreignField: 'id',
 as: 'venta'
}
```

El seller no puedo poner ventas.seller porque no esta dentro de un campo, esta dentro de un array --> $arrayelemat

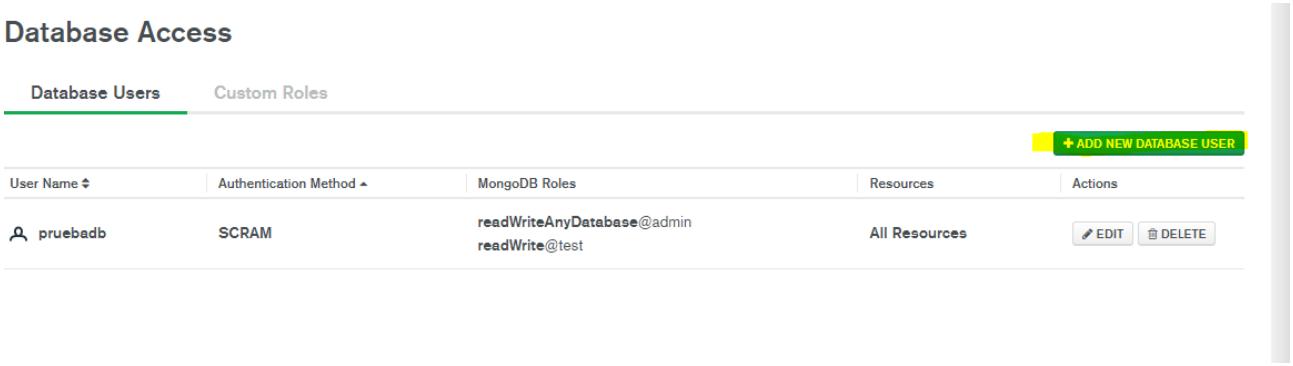
```
{
 name: 1,
 soldPrice:1,
 soldUnits:1,
 seller: "$seller.seller"
}
```

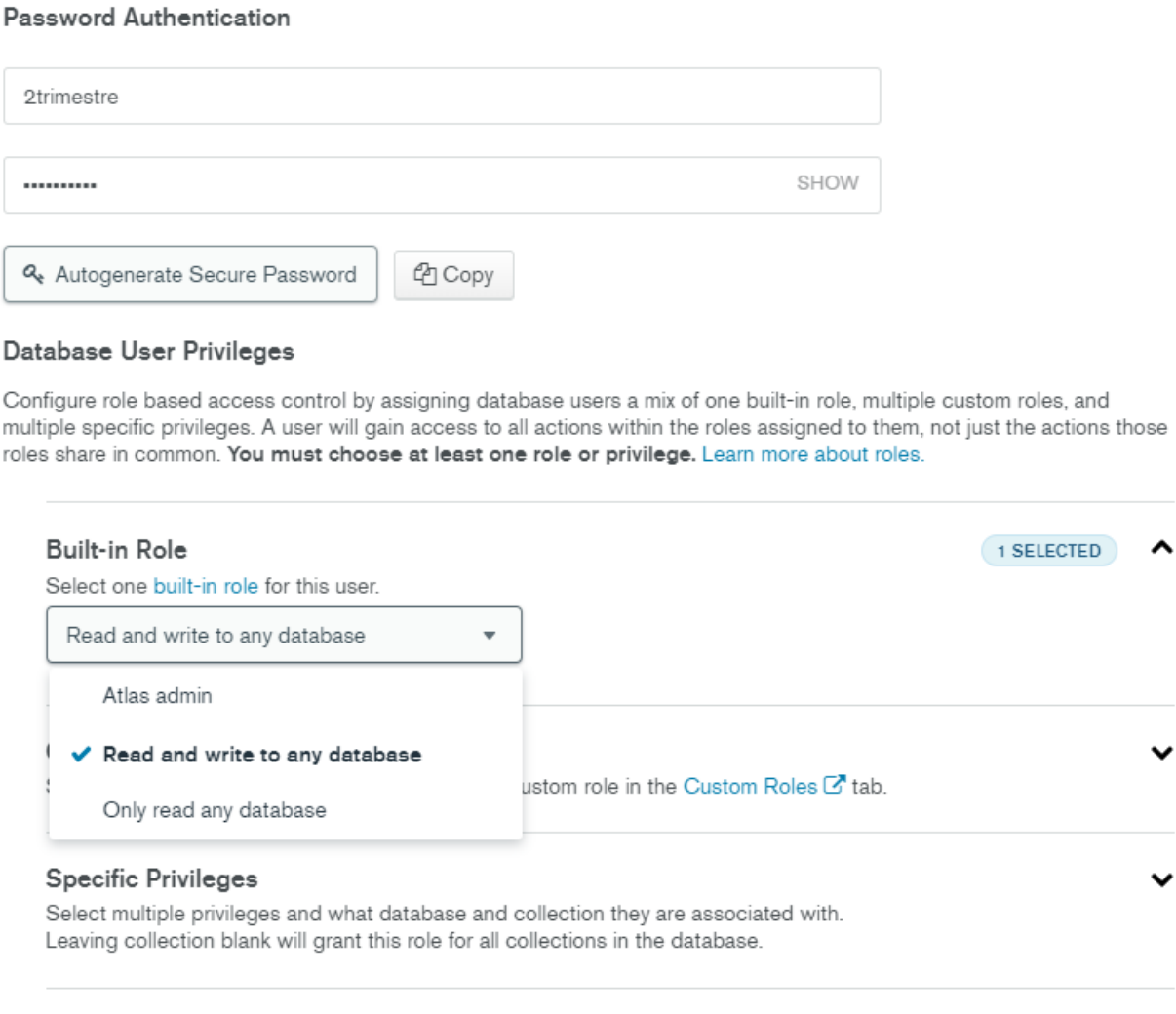
Aspectos técnicos

Entramos en **MongoAtlas** y lo conectamos con MongoCompass

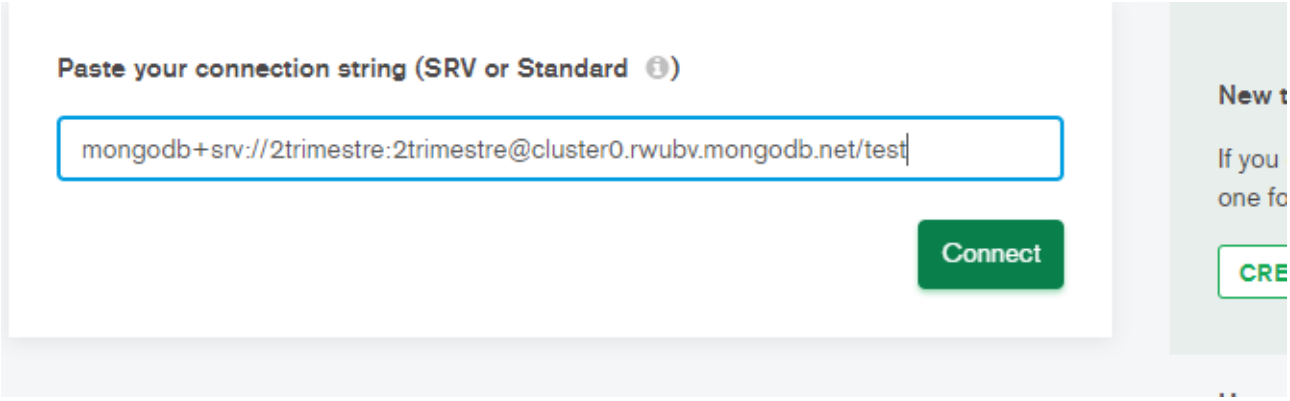


Creamos un nuevo usuario



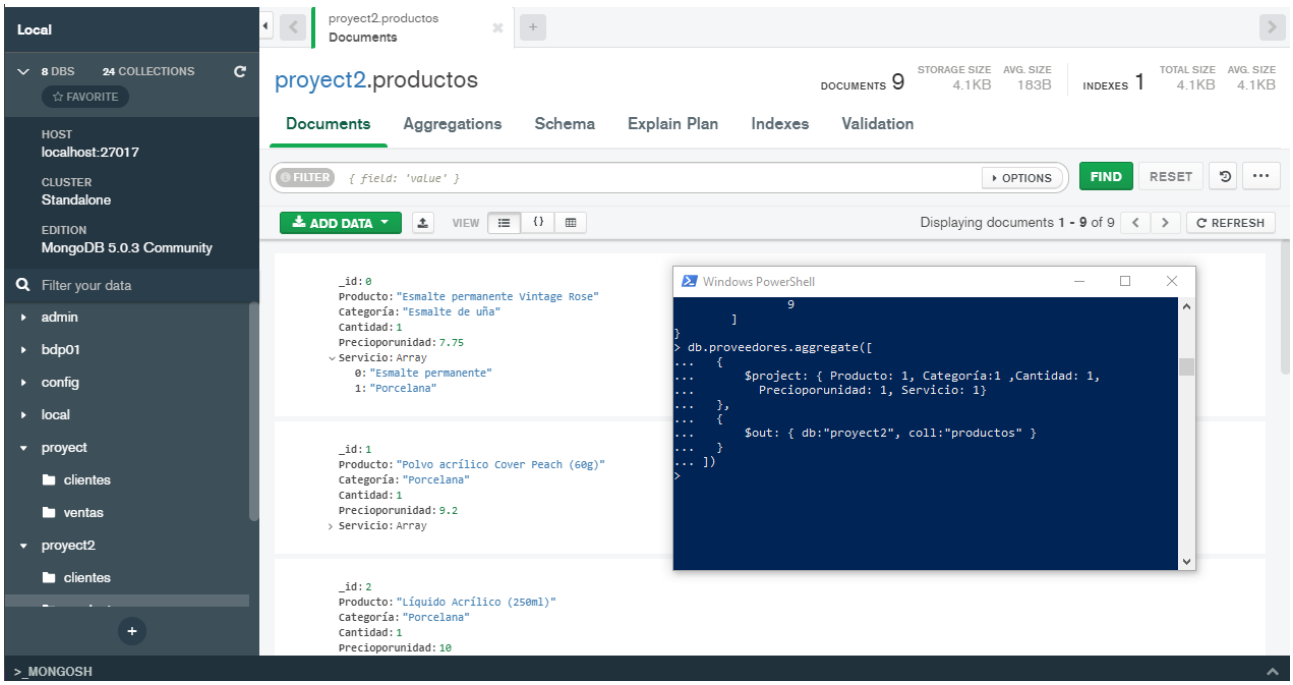


Y lo conectamos con Compass



«CONSULTAS»

Primero creamos una nueva colección a partir de la colección Proveedores usando aggregate



Hacemos la media de precio de todos los productos que compra.

```
> db.productos.aggregate([
...     {$group:
...         {
...             _id: "$producto",
...             MediaPrecio: {$avg: {$sum: "$Precioporunidad"}}
...         }
...
...
...     }
... ]).pretty()
{ "_id" : null, "MediaPrecio" : 6.515555555555555 }
>
```

Y unidades por cada categoría .

```
> db.proveedores.aggregate( [
...   {  $group: {  _id:"$Categoría",
...   Cantidad:{ $sum: 1 },
...   CantidadCategoría:{$sum: "$Cantidad"} }}
... ] )
{ "_id" : "Preparado de uña", "Cantidad" : 1, "CantidadCategoría" : 2 }
{ "_id" : "Porcelana", "Cantidad" : 4, "CantidadCategoría" : 4 }
{ "_id" : "Esmalte de uña", "Cantidad" : 1, "CantidadCategoría" : 1 }
{ "_id" : "Dibujo", "Cantidad" : 2, "CantidadCategoría" : 2 }
{ "_id" : "Diseño", "Cantidad" : 1, "CantidadCategoría" : 1 }
>
```

Numero de sesiones que se ha realizado cada persona.

```
>  db.clientes.aggregate(
...     [
...      {$group:
...              { _id:"$Nombre",
...                num_sesiones:
...                             {$sum:1}
...              }
...       }
...      ]
...    )
{ "_id" : "Pilar", "num_sesiones" : 2 }
{ "_id" : "Ángela", "num_sesiones" : 4 }
{ "_id" : "Carla", "num_sesiones" : 2 }
{ "_id" : "Angélica", "num_sesiones" : 1 }
{ "_id" : "Tita Ana", "num_sesiones" : 1 }
{ "_id" : "Brenda", "num_sesiones" : 1 }
{ "_id" : "Luisa del Pilar", "num_sesiones" : 1 }
{ "_id" : "Edu", "num_sesiones" : 2 }
{ "_id" : "Salud Escribano", "num_sesiones" : 1 }
{ "_id" : "Carolina Vecina", "num_sesiones" : 1 }
{ "_id" : "Fran", "num_sesiones" : 2 }
{ "_id" : "Tamara", "num_sesiones" : 3 }
{ "_id" : "Virginia", "num_sesiones" : 1 }
>
```

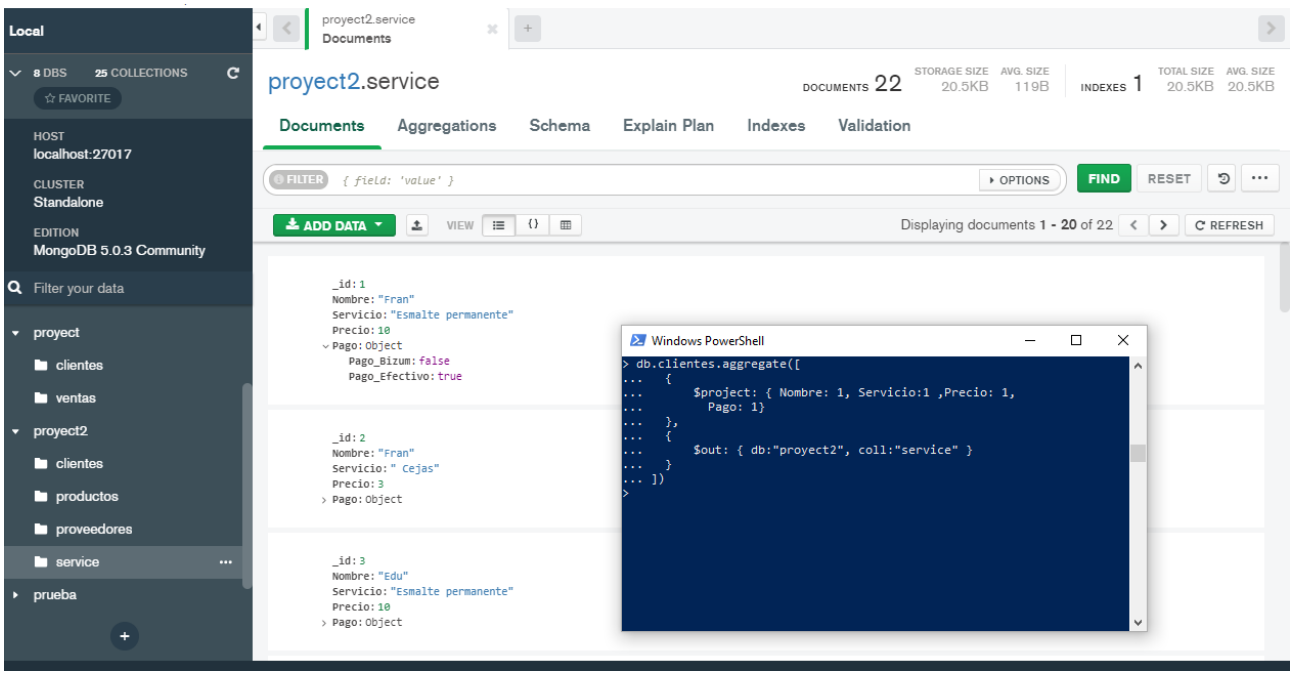
Separamos el array de servicios para facilitar mejor ver que servicio le corresponde a cada producto.

```
> db.proveedores.aggregate(
...   { $project : {
...     Producto: 1,
...       Categoría : 1 ,
...       Cantidad : 1 ,
...       Servicio : 1
...   }},
...   { $unwind : "$Servicio" }
... ).pretty()
{
        "_id" : 0,
        "Producto" : "Esmalte permanente Vintage Rose",
        "Categoría" : "Esmalte de uña",
        "Cantidad" : 1,
        "Servicio" : "Esmalte permanente"
}
{
        "_id" : 0,
        "Producto" : "Esmalte permanente Vintage Rose",
        "Categoría" : "Esmalte de uña",
        "Cantidad" : 1,
        "Servicio" : "Porcelana"
}
{
        "_id" : 1,
        "Producto" : "Polvo acrílico Cover Peach (60g)",
        "Categoría" : "Porcelana",
        "Cantidad" : 1,
        "Servicio" : "Porcelana"
}
{
        "_id" : 1,
        "Producto" : "Polvo acrílico Cover Peach (60g)",
        "Categoría" : "Porcelana",
        "Cantidad" : 1,
        "Servicio" : "Practica de Diseño"
}
{
        "_id" : 2,
        "Producto" : "Líquido Acrílico (250ml)",
        "Categoría" : "Porcelana",
        "Cantidad" : 1,
        "Servicio" : "Porcelana"
}
{
        "_id" : 2,
```

Esta consulta nos muestra el año el cual se compro cada producto.

```
> db.proveedores.aggregate(
...   [
...   { $sort: { fecha_compra: 1, Producto: 1 } },
...   {
...       $group:
...         {
...           _id: { day: { $dayOfYear: "$date"}, year: { $year: "$fecha_compra" } },
...           itemsSold: { $push:  { Producto: "$Producto", Cantidad: "$Cantidad" } }
...         }
...     }
...   ]
... ).pretty()
{
        "_id" : {
                "day" : null,
                "year" : 2020
        },
        "itemsSold" : [
                {
                        "Producto" : "Polvo Acrílico Butterfly (10gr)",
                        "Cantidad" : 1
                },
                {
                        "Producto" : "Esmalte permanente Vintage Rose",
                        "Cantidad" : 1
                },
                {
                        "Producto" : "Polvo acrílico Cover Peach (60g)",
                        "Cantidad" : 1
                }
        ]
}
{
        "_id" : {
                "day" : null,
                "year" : 2022
        },
        "itemsSold" : [
```

Creamos una nueva colección a partir de la colección clientes para luego hacer un $lookup con la colección productos.



Con esta consulta he querido reflejar los materiales que tenemos disponibles para utilizarlos en cada cliente en cuanto al servicio prestado. Y si ha pagado ese servicio en efectivo o bizum.

```
db.service.aggregate([
 {      $lookup:
  {          from: "productos",
   localField: "Servicio.id",
    foreignField: "id",
     as: "Producto"        }   } ]).pretty()
```

```
{
        "_id" : 20,
        "Nombre" : "Angélica",
        "Servicio" : "Esmalte permanente",
        "Precio" : 10,
        "Pago" : {
                "Pago_Bizum" : false,
                "Pago_Efectivo" : true
        },
        "Producto" : [
                {
                        "_id" : 0,
                        "Producto" : "Esmalte permanente Vintage Rose",
                        "Categoría" : "Esmalte de uña",
                        "Cantidad" : 1,
                        "Precioporunidad" : 7.75,
                        "Servicio" : [
                                "Esmalte permanente",
                                "Porcelana"
                        ]
                },
                {
                        "_id" : 1,
                        "Producto" : "Polvo acrílico Cover Peach (60g)",
                        "Categoría" : "Porcelana",
                        "Cantidad" : 1,
                        "Precioporunidad" : 9.2,
                        "Servicio" : [
                                "Porcelana",
                                "Practica de Diseño"
                        ]
                },
                {
```

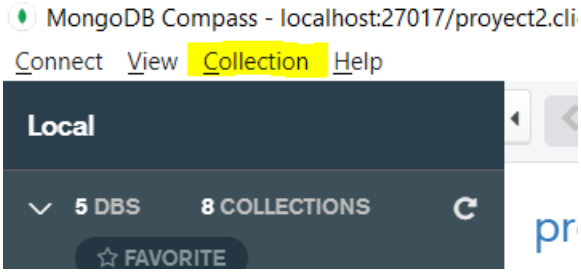

MONGOIMPORT & MONGOEXPORT

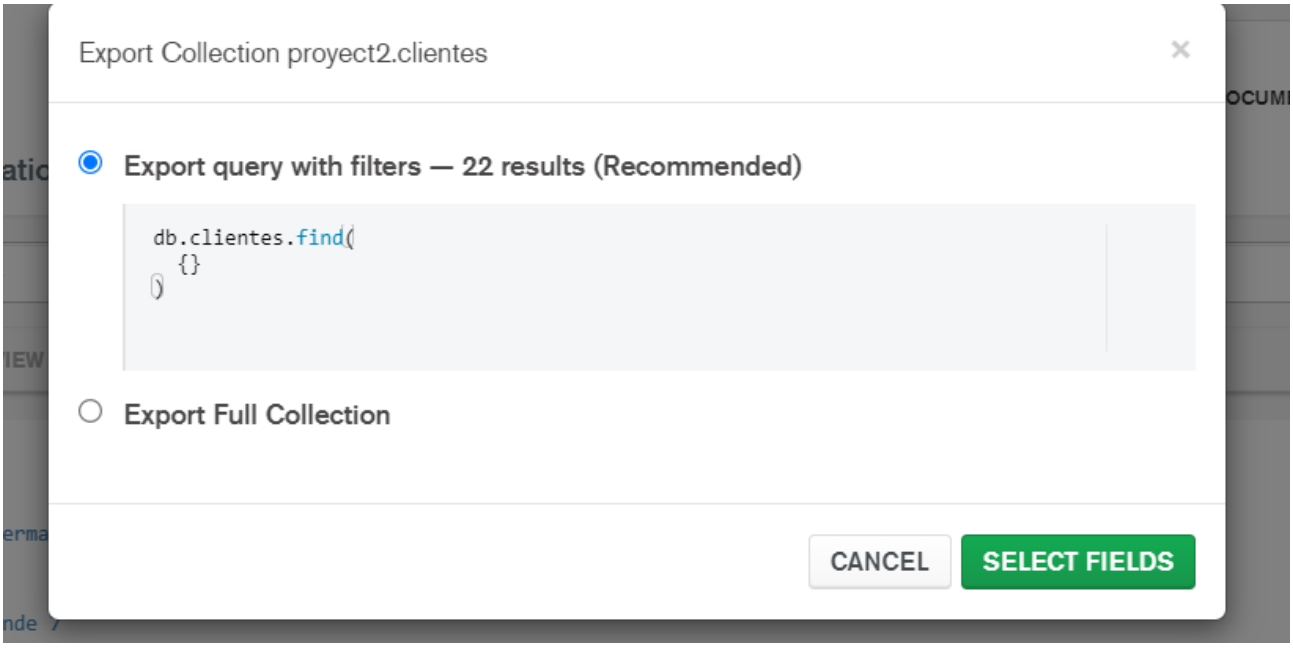
Para exportar e importar los datos podemos usar esta sintaxis o desde Compass en modo interactivo.

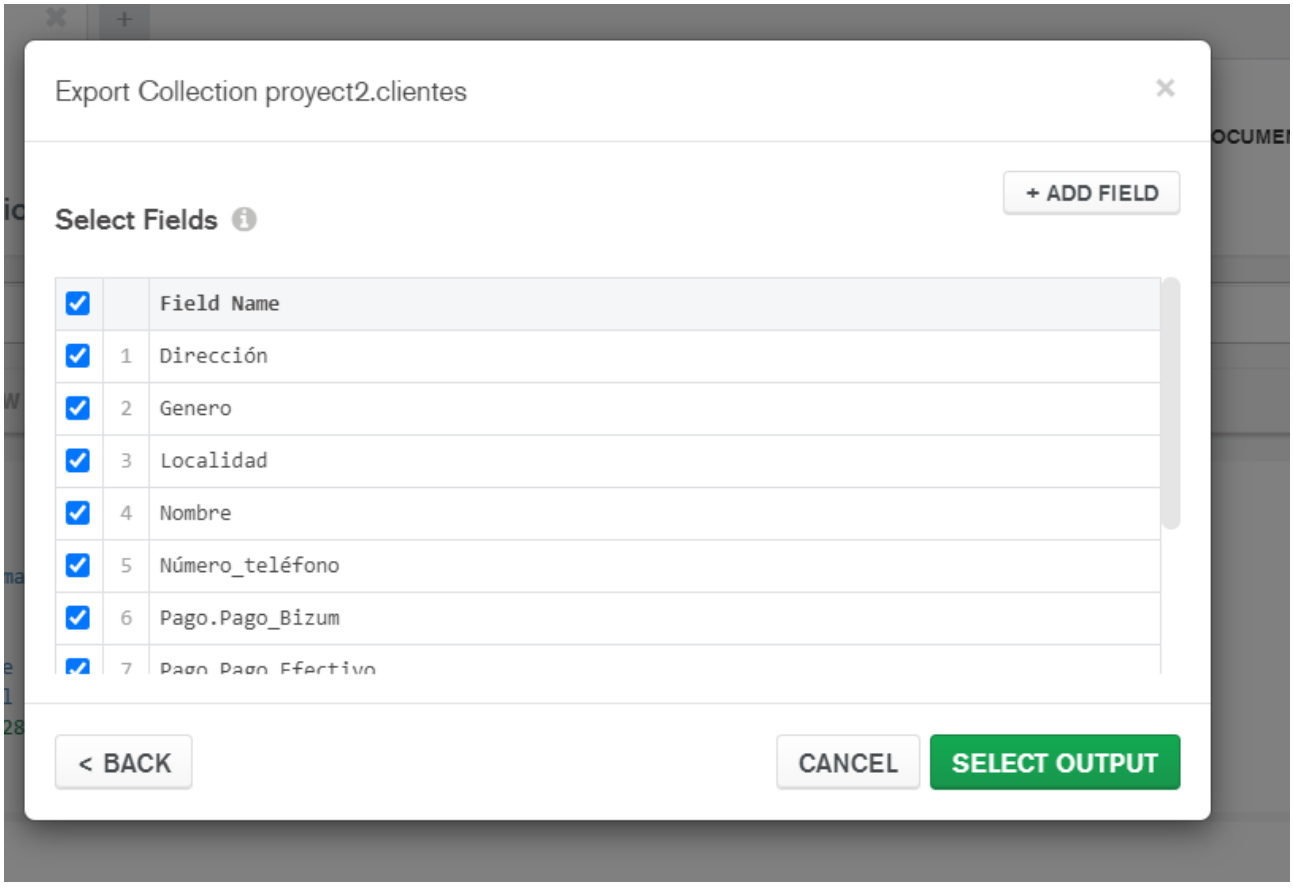
mongoexport --collection clientes --db test --type csv --fields nombre,precio --out C:\Users\usuario1\documents\basededatos\proyecto05\src fichero.csv

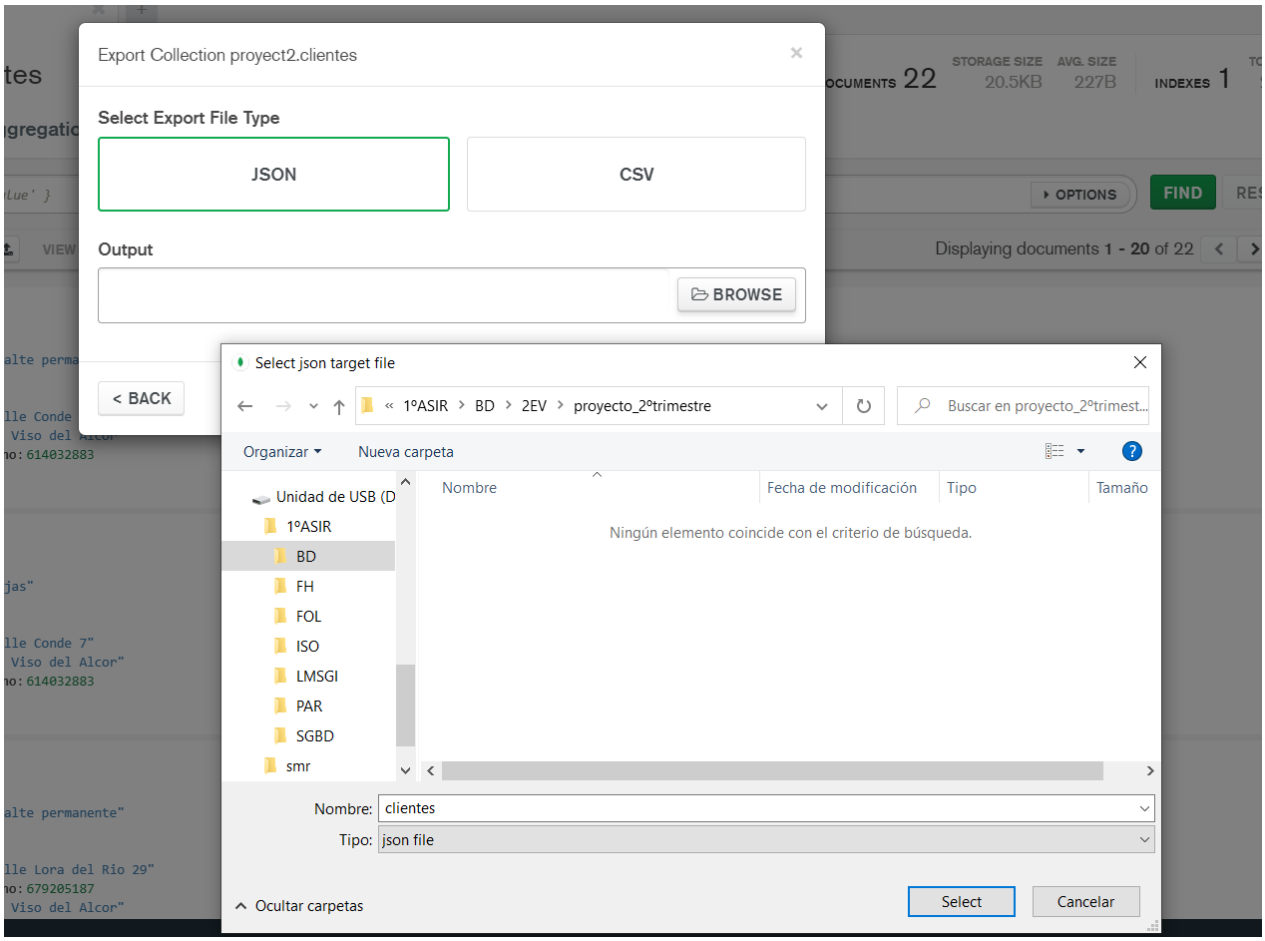
mongoexport /d test /c collection /type json /o collection.json

MONGO EXPORT



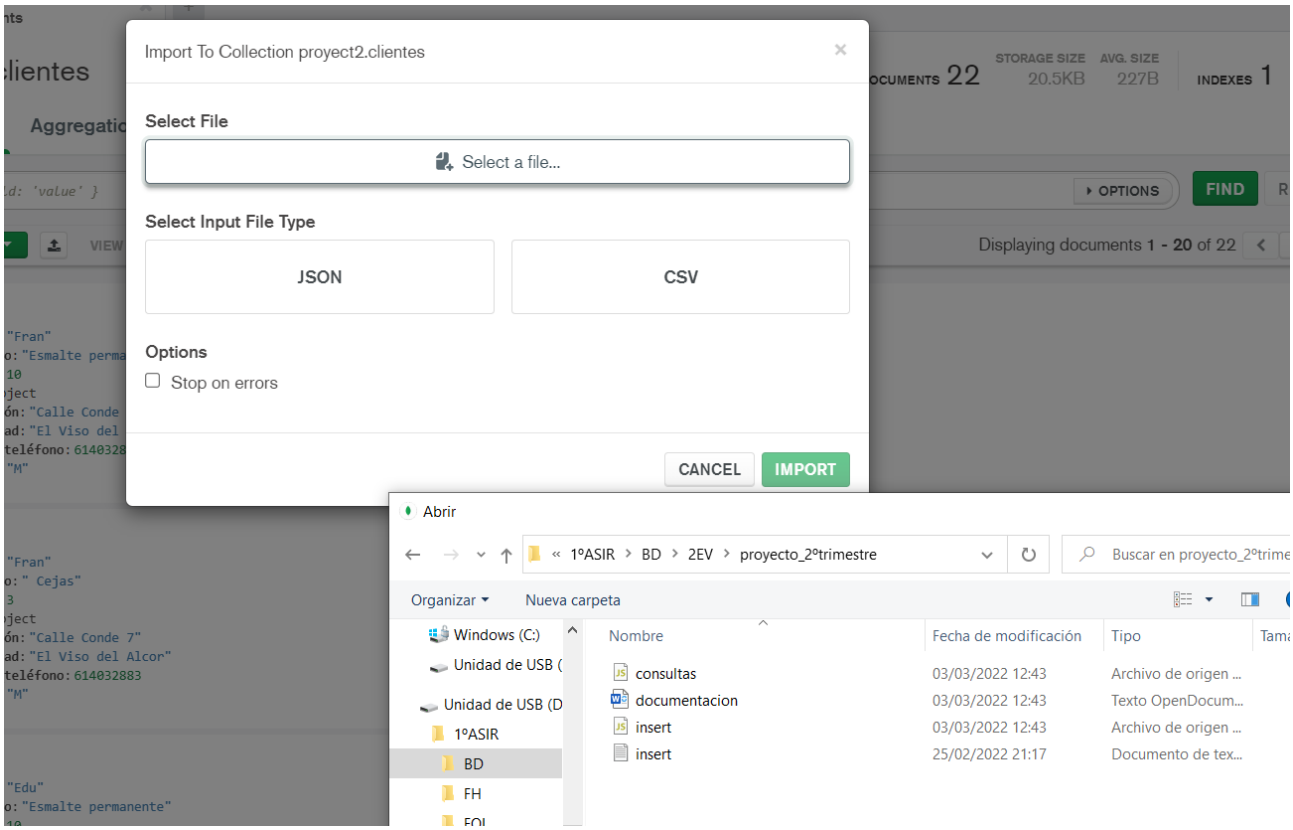

Export Collection proyect2.clientes
Export query with filters — 22 results (Recommended)
db.clientes.find(
{}
)
Export Full Collection
CANCEL
SELECT FIELDS


Export Collection proyect2.clientes
Select Fields
+ ADD FIELD
Field Name
1 Dirección
2 Genero
3 Localidad
4 Nombre
5 Número_teléfono
6 Pago.Pago_Bizum
7 Pago.Pago_Efectivo
< BACK
CANCEL
SELECT OUTPUT



MONGO IMPORT

Importamos una colección ya creada a una base de datos.



Formato CSV Son valores separados por comas, se usan normalmente para intercambiar datos tabulares entre sistemas en texto sin formato. Normalmente contienen una fila de encabezado que proporciona los nombres de columna para los datos, pero estos se consideran semiestructurados. CSV no pueden representar de forma natural datos jerárquicos o relacionales. Las relaciones de datos normalmente se controlan con varios archivos CSV, donde las claves externas se almacenan en columnas de uno o más archivos, pero las relaciones entre esos archivos no se expresan con el propio formato. Los archivos en formato CSV pueden usar otros delimitadores además de las comas, por ejemplo, tabulaciones o espacios.

Formato JSON Notación de objetos JavaScript, se representan como pares de clave y valor en un formato semiestructurado. JSON se compara a menudo con XML, ya que ambos son capaces de almacenar los datos en un formato jerárquico con los datos secundarios representados alineados con su elemento primario. Ambos son autodescriptivos y legibles, pero los documentos JSON tienden a ser mucho más pequeños, lo que ha llevado a su uso popular en el intercambio de datos en línea, sobre todo con la llegada de los servicios web basados en REST.