

Proyecto Mongo y Visual Studio Code.

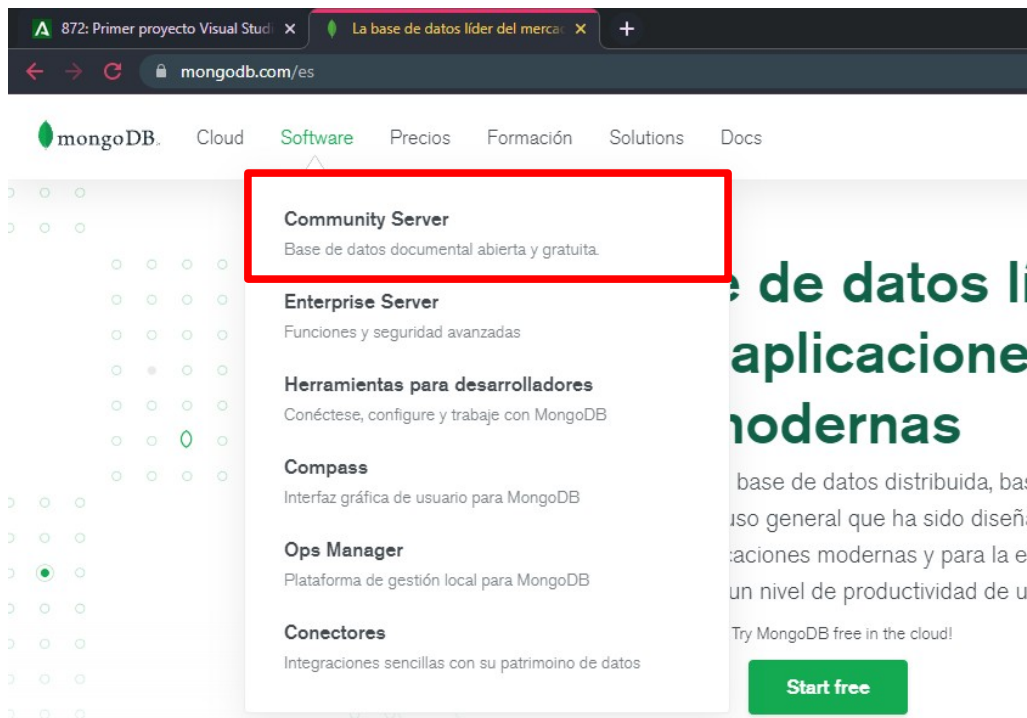
-Gestión de bases de datos no sql-

Índice

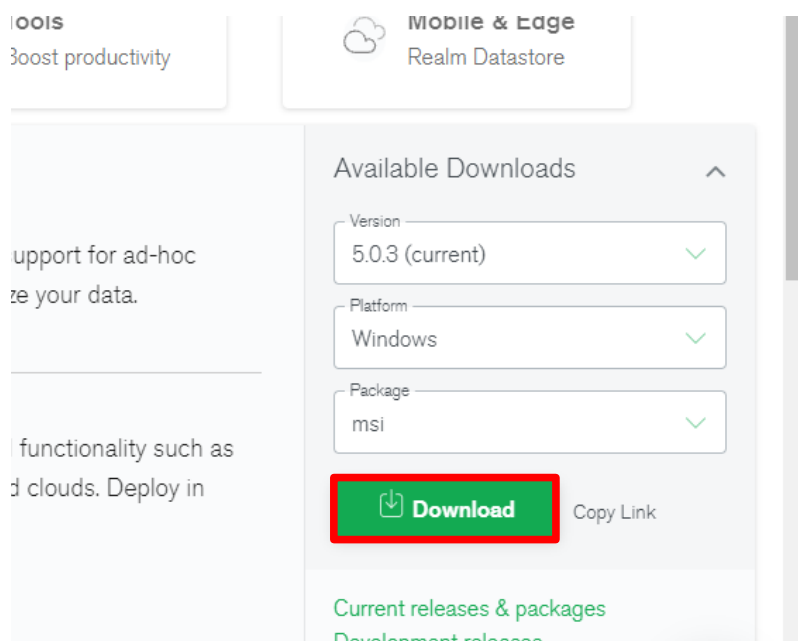
1.Instalación MongoDB.....	3
2.Instalación de Visual Studio Code.....	8
3.Trabajando con la base de datos.....	9
3.1.Creación de la base de datos.....	9
3.2.Consultas en la base de datos.....	13

1.Instalación MongoDB.

Para instalar MongoDB debemos dirigirnos a su página oficial, una vez aquí en el apartado de software pulsamos la opción de Community Server.



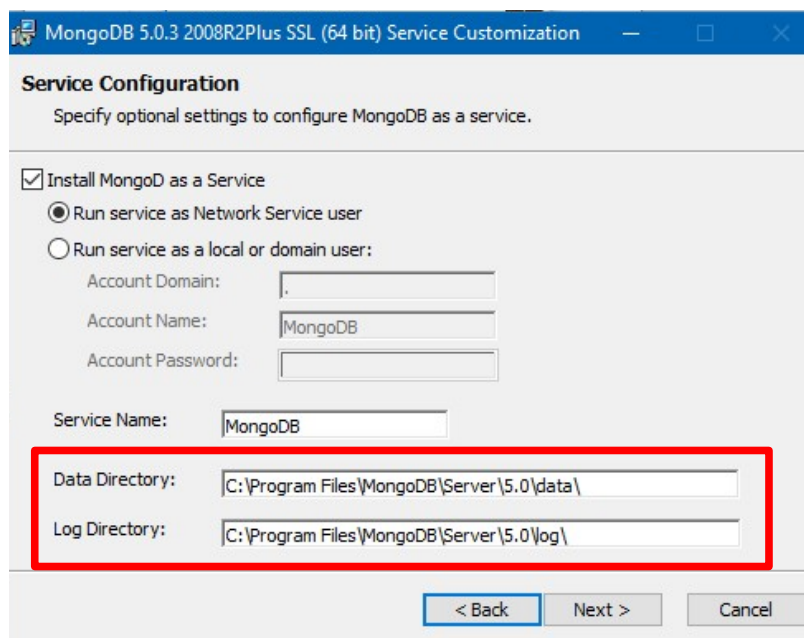
Dentro del apartado de Community Server, pulsaremos el botón de descargar de la derecha.



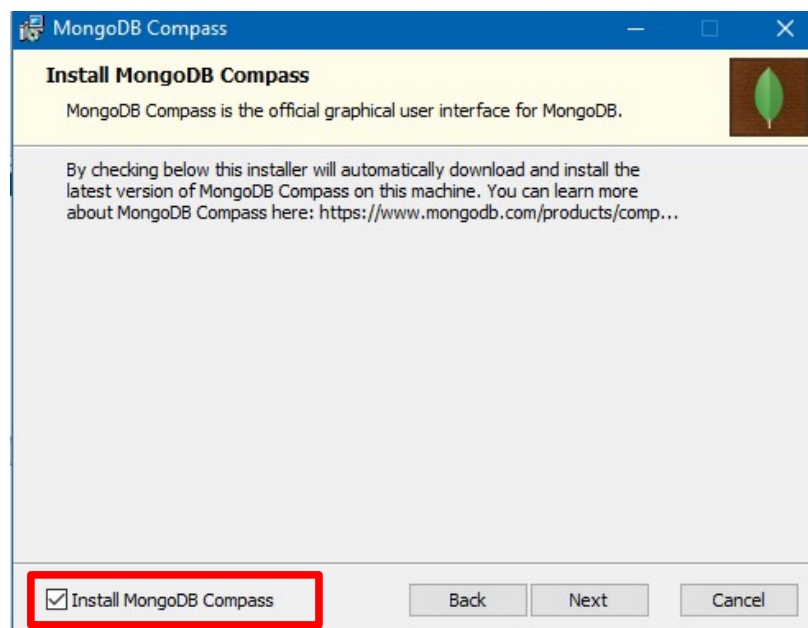
Ahora ejecutamos el instalador que nos hemos descargado y comenzamos con la instalación de Mongo. Al arrancar debemos aceptar los términos de licencia y pulsar en instalación completa.



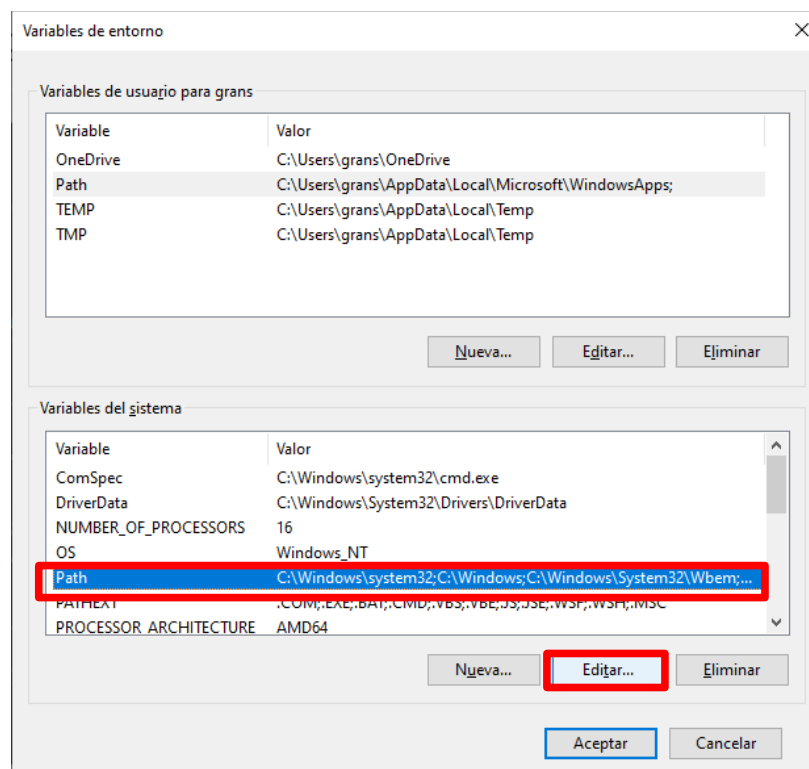
En la siguiente pantalla elegimos donde instalar mongo y dejamos los demás valores por defecto.



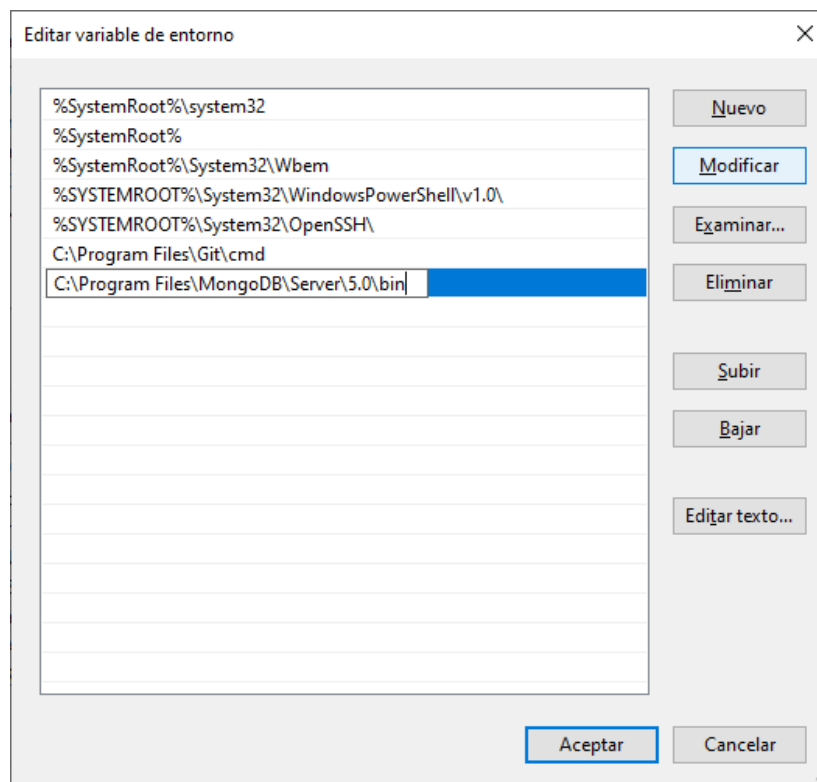
Ahora instalamos Mongo Compass que es el entorno gráfico de Mongo, seleccionamos la opción de abajo e instalamos.



Ahora comenzaría la instalación del programa, una vez finalizada, debemos añadir mongo al path para que lo reconozca la consola. Para ello nos dirigiremos a las variables del entorno y editamos la variable path.



Una vez aquí pulsamos en nuevo y pegamos la ruta donde tengamos instalado mongo.

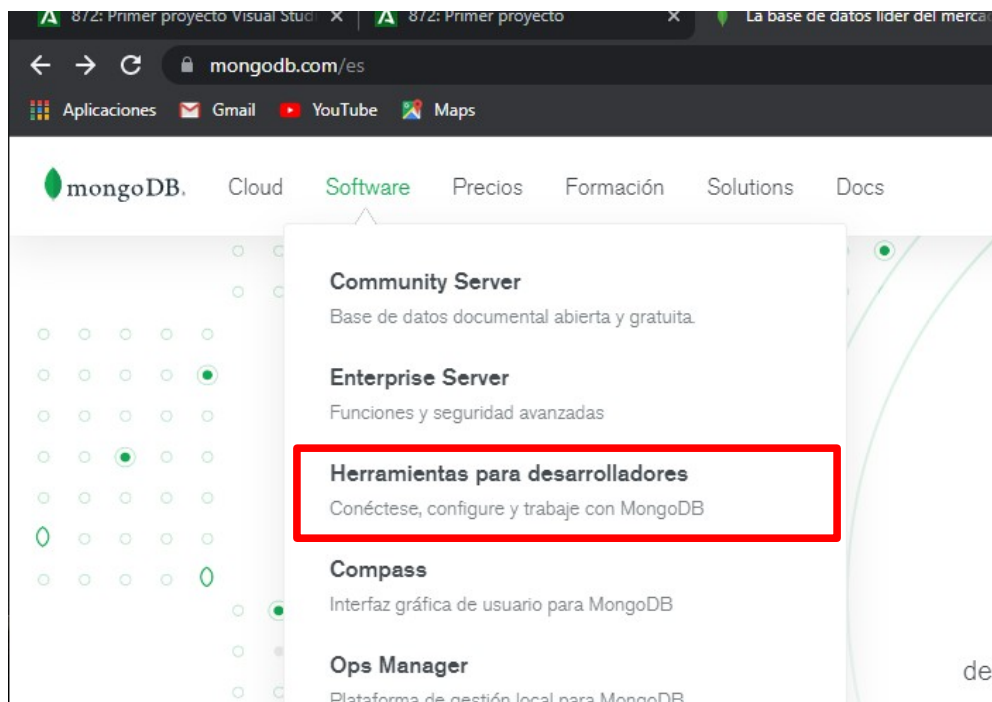


Aceptamos y ahora al ejecutar mongo en una Power Shell nos debería funcionar.

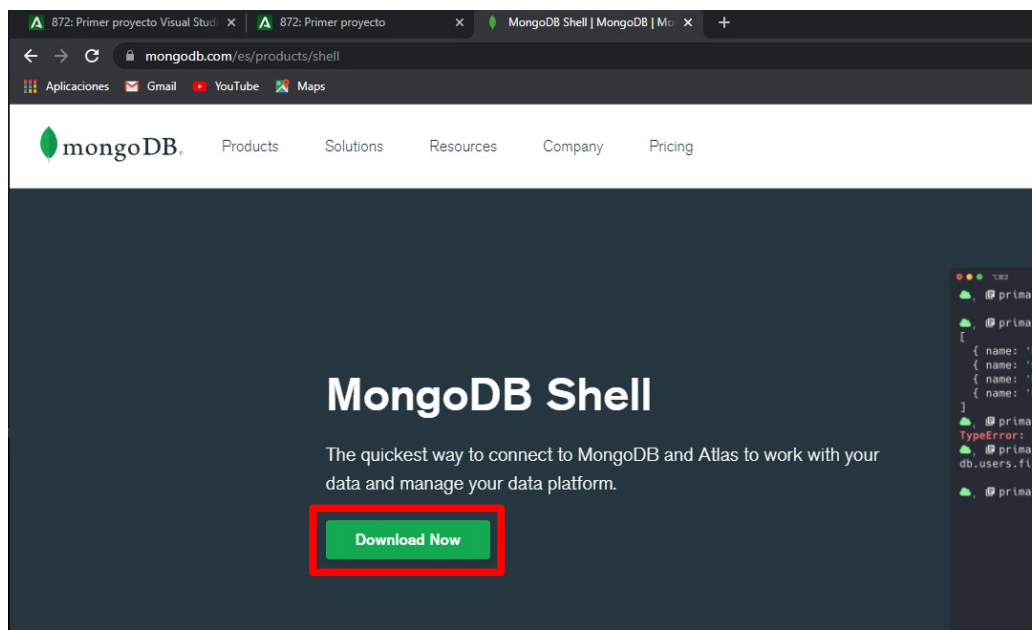
```
Selecionar Windows PowerShell
PS C:\Users\grans> mongo --version
MongoDB shell version v5.0.3
Build Info: {
  "version": "5.0.3",
  "gitVersion": "657fea5a61a74d7a79df7aff8e4bcf0bc742b748",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
PS C:\Users\grans>
```

Como vemos, mongo funciona perfectamente.

Por último vamos a instalar una shell para mongo mas moderna llamada mongosh, para ello en su página web nos vamos a Software y a Herramientas para desarrolladores.



En este apartado nos vamos a MongoDB Shell y la descargamos.

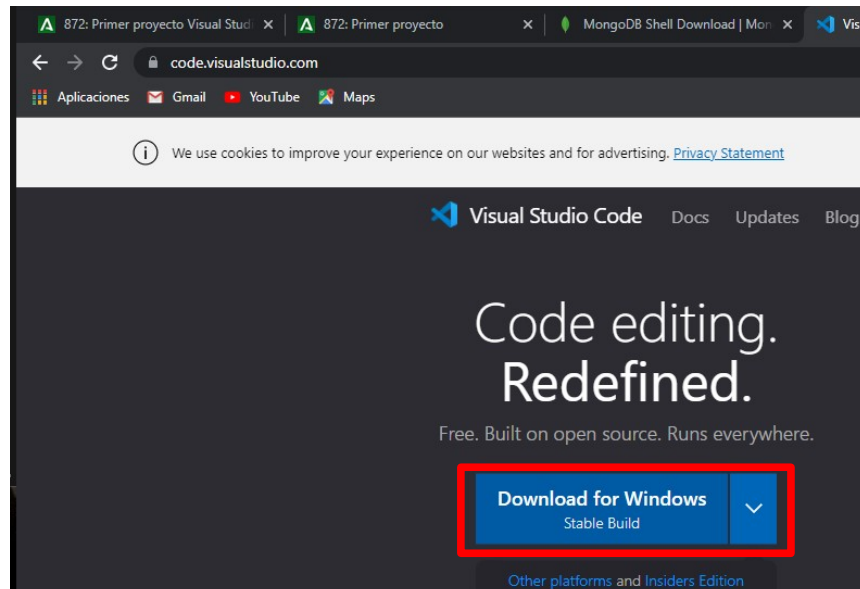


Ejecutamos el instalador, elegimos la ruta para instalar e instalamos la nueva shell, cuando la instalemos nos dirigimos a una Power Shell y ejecutamos el comando mongosh, si no funciona debemos añadirlo al path al igual que mongo.

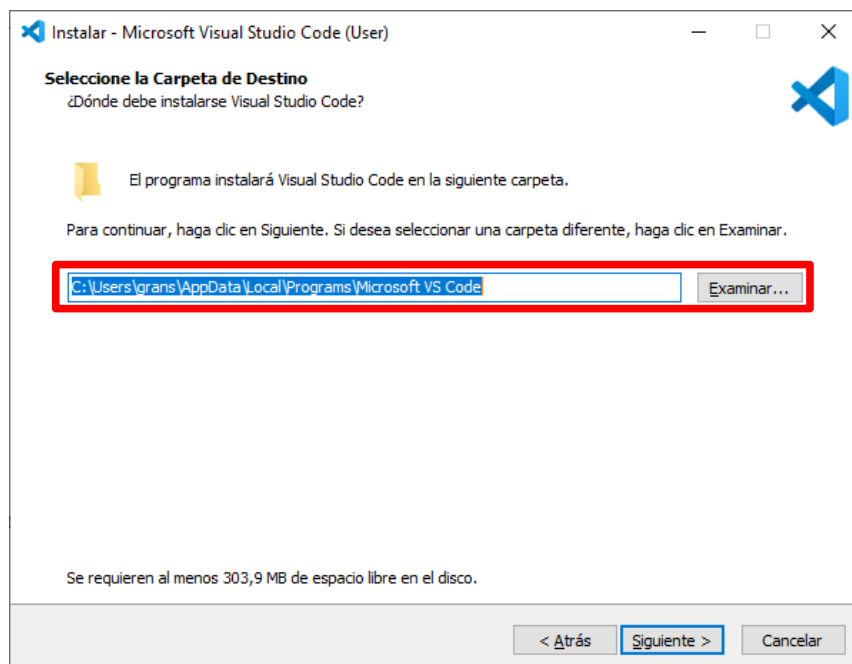
Esta shell tiene nuevas funcionalidades pero todavía no esta acabada y da fallos por lo que yo usaré la shell normal de mongo.

2.Instalación de Visual Studio Code.

Para instalar Visual Studio Code nos vamos a la página oficial y pulsamos el botón de descargar.



Una vez descargado ejecutamos el instalador, aceptamos los términos y elegimos la ruta de instalación.



Pulsamos en instalar y esperamos a que la instalación finalice.

Ya tenemos instalado tanto mongo como visual studio por lo que podremos empezar con el proyecto.

3.Trabajando con la base de datos.

Para este proyecto he decidido crear una base de datos basada en un taller, en el que se arreglarán coches y motos, la bases de datos constará de dos colecciones y usaremos todos los comandos vistos en clase, métodos inserts y operadores query.

3.1.Creación de la base de datos.

Nuestra base de datos se llamara tallerdb, que contendrá dos colecciones, una de motos y otra de coches.

Para crear la base de datos ejecutamos mongo en una Power Shell y usamos el comando use tallerdb , de esta forma usaremos la base de datos tallerdb que al no existir la creara, comprobamos después que estamos en esa base de datos con el comando db.

```
> use tallerdb
switched to db tallerdb
> db
tallerdb
> █
```

Como vemos la base de datos se ha creado correctamente, ahora vamos a crear las dos colecciones.

Para crear la colección moto vamos a insertar un documento dentro de ella con un método insertOne.

```
> db.motos.insertOne({marca: "ducati", modelo: "scrambler", color: "rojo", dimensiones: {altuta: 1, ancho: 0.78, largo: 2}, matricula: "1764HFK"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("6161b82843692ee748226c99")
}
> █
```

Esos son los datos que vamos a guardar de cada moto, como vemos con insertOne insertamos datos en un solo documento.

Realizaremos el mismo proceso con los coches.

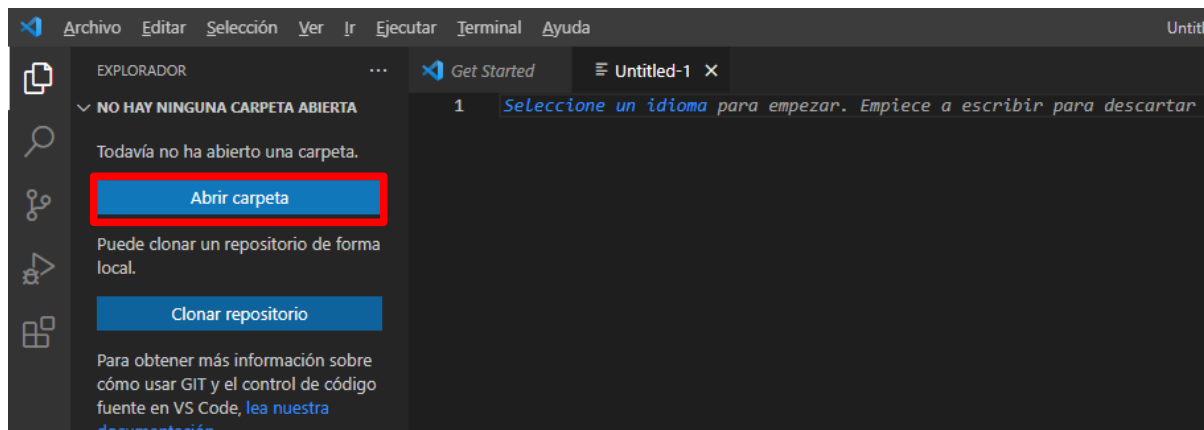
```
> db.coches.insertOne({marca: "opel", modelo: "corsa", color: "gris", dimensiones: {altura: 1.45, ancho: 1.78, largo: 3.5}, matricula: "1890SDF"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("6161b94e43692ee748226c9a")
}
```

Una vez añadidos los dos documentos, vamos a hacer un show collections para comprobar que las colecciones se hayan creado correctamente.

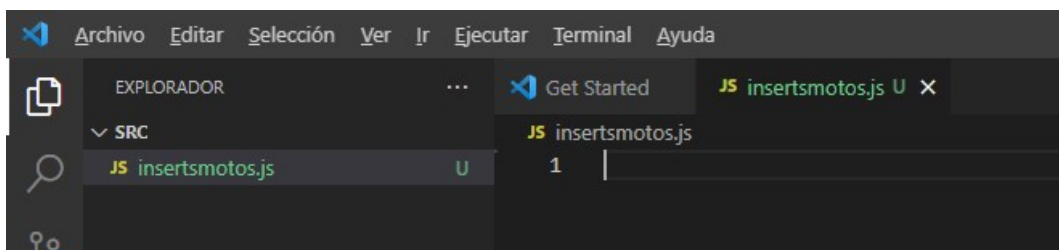
```
> show collections
coches
motos
>
```

Como vemos están creadas las dos colecciones.

Ahora vamos a añadir datos mediante el uso de ficheros json, para ello nos vamos a visual studio code y pulsamos en nuevo proyecto y seleccionamos la opción abrir carpeta, debemos abrir la carpeta src de nuestro proyecto ya que es donde guardaremos todos los ficheros.

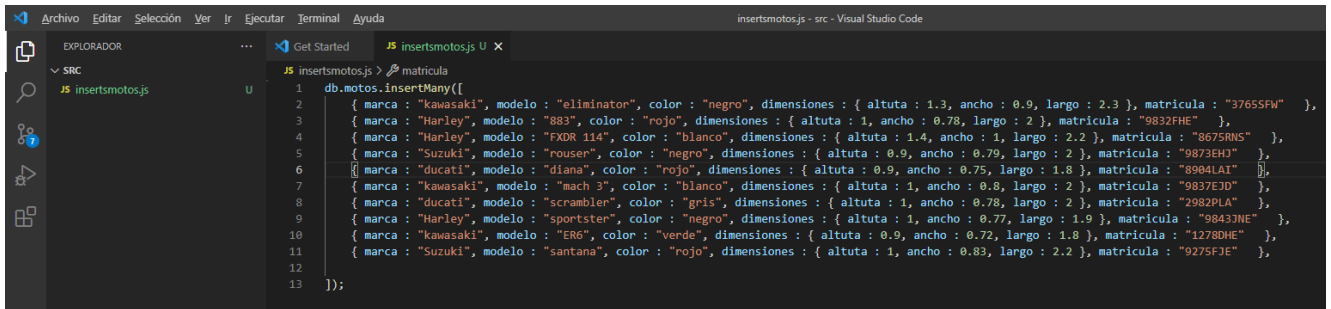


Una vez abierta la carpeta creamos un nuevo archivo, el primer archivo que creará será para insertar datos en la colección de motos, por lo que mi fichero se llamará insertsmotos.js.



Ahora debemos añadir los comandos para insertar datos a la base de datos.

Como en el paso anterior añadimos datos usando el comando insertOne, que nos permite insertar un solo documento por comando, en los ficheros usaremos el comando insertMany, que nos permite añadir un array de documentos por lo que podremos añadir mucha mas información a la base de datos.



```

1 db.motos.insertMany([
2   { marca : "kawasaki", modelo : "eliminator", color : "negro", dimensiones : { altura : 1.3, ancho : 0.9, largo : 2.3 }, matricula : "37655FW" },
3   { marca : "Harley", modelo : "883", color : "rojo", dimensiones : { altura : 1, ancho : 0.78, largo : 2 }, matricula : "9832FHE" },
4   { marca : "Harley", modelo : "FXDR 114", color : "blanco", dimensiones : { altura : 1.4, ancho : 1, largo : 2.2 }, matricula : "8675RNS" },
5   { marca : "Suzuki", modelo : "rouser", color : "negro", dimensiones : { altura : 0.9, ancho : 0.70, largo : 2 }, matricula : "9873EHJ" },
6   { marca : "ducati", modelo : "diana", color : "rojo", dimensiones : { altura : 0.9, ancho : 0.75, largo : 1.8 }, matricula : "8904LAI" },
7   { marca : "kawasaki", modelo : "mach 3", color : "blanco", dimensiones : { altura : 1, ancho : 0.8, largo : 2 }, matricula : "9837E3D" },
8   { marca : "ducati", modelo : "scrambler", color : "gris", dimensiones : { altura : 1, ancho : 0.78, largo : 2 }, matricula : "2982PLA" },
9   { marca : "Harley", modelo : "sportster", color : "negro", dimensiones : { altura : 1, ancho : 0.77, largo : 1.9 }, matricula : "9843JNE" },
10  { marca : "kawasaki", modelo : "ER6", color : "verde", dimensiones : { altura : 0.9, ancho : 0.72, largo : 1.8 }, matricula : "1278DHE" },
11  { marca : "Suzuki", modelo : "santana", color : "rojo", dimensiones : { altura : 1, ancho : 0.83, largo : 2.2 }, matricula : "9275FJE" },
12
13 ]);
  
```

Como vemos esta es la estructura de nuestro fichero, es un conjunto de documentos similares al primero que insertamos, separados por comas y dentro de un corchete. Guardamos este fichero y lo ejecutamos dentro de nuestra base de datos con el comando load, para eso antes de ejecutar mongo debemos situarnos dentro de la carpeta src.

```

PS C:\Users\grans> cd F:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto02\src\
PS F:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto02\src> ls

Directorio: F:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto02\src

Mode                LastWriteTime         Length Name
----                -
-a----            10/10/2021   12:04           1485 insertsmotos.js

PS F:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto02\src> mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("2326f6c5-d580-4f47-a933-18d96828a8c6") }
MongoDB server version: 5.0.3
=====
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/
=====
The server generated these startup warnings when booting:
  2021-10-10T11:18:28.259+02:00: Access control is not enabled for the database. Read and write access
s to data and configuration is unrestricted
-----
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----
>
  
```

Ahora accedemos a nuestra base de datos y cargamos el fichero.

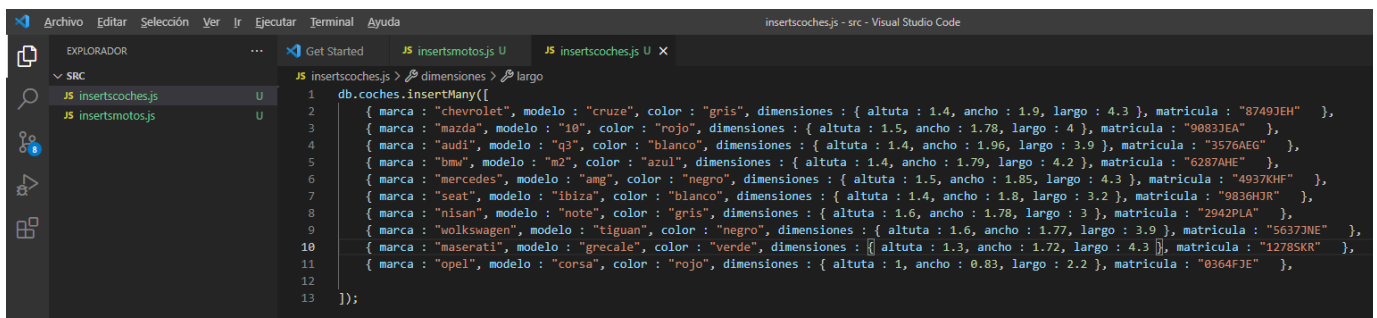
```
> use tallerdb
switched to db tallerdb
> show collections
coches
motos
> load("insertsmotos.js")
true
>
```

Ejecutamos un find en la colección motos para comprobar que los datos se han guardado correctamente.

```
> db.motos.find()
{ "_id" : ObjectId("6161b82843692ee748226c99"), "marca" : "ducati", "modelo" : "scrambler", "color" : "rojo", "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "1764HFK" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a6"), "marca" : "kawasaki", "modelo" : "eliminator", "color" : "negro", "dimensiones" : { "altuta" : 1.3, "ancho" : 0.9, "largo" : 2.3 }, "matricula" : "3765SFW" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a7"), "marca" : "Harley", "modelo" : "883", "color" : "rojo", "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "9832FHE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a8"), "marca" : "Harley", "modelo" : "FXDR 114", "color" : "blanco", "dimensiones" : { "altuta" : 1.4, "ancho" : 1, "largo" : 2.2 }, "matricula" : "8675RNS" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a9"), "marca" : "Suzuki", "modelo" : "rouser", "color" : "negro", "dimensiones" : { "altuta" : 0.9, "ancho" : 0.79, "largo" : 2 }, "matricula" : "9873EHJ" }
{ "_id" : ObjectId("6162bc51f07633cad402a5aa"), "marca" : "ducati", "modelo" : "diana", "color" : "rojo", "dimensiones" : { "altuta" : 0.9, "ancho" : 0.75, "largo" : 1.8 }, "matricula" : "8904LAI" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ab"), "marca" : "kawasaki", "modelo" : "mach 3", "color" : "blanco", "dimensiones" : { "altuta" : 1, "ancho" : 0.8, "largo" : 2 }, "matricula" : "9837EJD" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ac"), "marca" : "ducati", "modelo" : "scrambler", "color" : "gris", "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "2982PLA" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ad"), "marca" : "Harley", "modelo" : "sportster", "color" : "negro", "dimensiones" : { "altuta" : 1, "ancho" : 0.77, "largo" : 1.9 }, "matricula" : "9843JNE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ae"), "marca" : "kawasaki", "modelo" : "ER6", "color" : "verde", "dimensiones" : { "altuta" : 0.9, "ancho" : 0.72, "largo" : 1.8 }, "matricula" : "1278DHE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5af"), "marca" : "Suzuki", "modelo" : "santana", "color" : "rojo", "dimensiones" : { "altuta" : 1, "ancho" : 0.83, "largo" : 2.2 }, "matricula" : "9275FJE" }
>
```

Como vemos los documentos se han añadido correctamente a la base de datos.

Realizamos la misma acción con la colección coches.



```
1 db.coches.insertMany([
2   { marca : "chevrolet", modelo : "cruze", color : "gris", dimensiones : { altuta : 1.4, ancho : 1.9, largo : 4.3 }, matricula : "8749JEH" },
3   { marca : "mazda", modelo : "10", color : "rojo", dimensiones : { altuta : 1.5, ancho : 1.78, largo : 4 }, matricula : "9083JEA" },
4   { marca : "audi", modelo : "q3", color : "blanco", dimensiones : { altuta : 1.4, ancho : 1.96, largo : 3.9 }, matricula : "3576AEG" },
5   { marca : "bmw", modelo : "m2", color : "azul", dimensiones : { altuta : 1.4, ancho : 1.79, largo : 4.2 }, matricula : "6287AHE" },
6   { marca : "mercedes", modelo : "amg", color : "negro", dimensiones : { altuta : 1.5, ancho : 1.85, largo : 4.3 }, matricula : "4937KHF" },
7   { marca : "seat", modelo : "ibiza", color : "blanco", dimensiones : { altuta : 1.4, ancho : 1.8, largo : 3.2 }, matricula : "9836HJR" },
8   { marca : "nissan", modelo : "note", color : "gris", dimensiones : { altuta : 1.6, ancho : 1.78, largo : 3 }, matricula : "2942PLA" },
9   { marca : "volkswagen", modelo : "tiguan", color : "negro", dimensiones : { altuta : 1.6, ancho : 1.77, largo : 3.9 }, matricula : "5637JNE" },
10  { marca : "maserati", modelo : "grecale", color : "verde", dimensiones : { altuta : 1.3, ancho : 1.72, largo : 4.3 }, matricula : "1278SKR" },
11  { marca : "opel", modelo : "corsa", color : "rojo", dimensiones : { altuta : 1, ancho : 0.83, largo : 2.2 }, matricula : "0364FJE" },
12
13 ]);
```

Insertamos este fichero en la base de datos y comprobamos que todo se ha cargado correctamente.

```
> load("insertscoches.js")
true
> db.coches.find()
{ "_id" : ObjectId("6161b94e43692ee748226c9a"), "marca" : "opel", "modelo" : "corsa", "color" : "gris", "dimensiones" : { "altuta" : 1.45, "ancho" : 1.78, "largo" : 3.5 }, "matricula" : "1890SDF" }
{ "_id" : ObjectId("6162c075f07633cad402a5b0"), "marca" : "chevrolet", "modelo" : "cruze", "color" : "gris", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.9, "largo" : 4.3 }, "matricula" : "8749JEH" }
{ "_id" : ObjectId("6162c075f07633cad402a5b1"), "marca" : "mazda", "modelo" : "i10", "color" : "rojo", "dimensiones" : { "altuta" : 1.5, "ancho" : 1.78, "largo" : 4 }, "matricula" : "9083JEA" }
{ "_id" : ObjectId("6162c075f07633cad402a5b2"), "marca" : "audi", "modelo" : "q3", "color" : "blanco", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.96, "largo" : 3.9 }, "matricula" : "3576AEG" }
{ "_id" : ObjectId("6162c075f07633cad402a5b3"), "marca" : "bmw", "modelo" : "m2", "color" : "azul", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.79, "largo" : 4.2 }, "matricula" : "6287AHE" }
{ "_id" : ObjectId("6162c075f07633cad402a5b4"), "marca" : "mercedes", "modelo" : "amg", "color" : "negro", "dimensiones" : { "altuta" : 1.5, "ancho" : 1.85, "largo" : 4.3 }, "matricula" : "4937KHF" }
{ "_id" : ObjectId("6162c075f07633cad402a5b5"), "marca" : "seat", "modelo" : "ibiza", "color" : "blanco", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.8, "largo" : 3.2 }, "matricula" : "9836HJR" }
{ "_id" : ObjectId("6162c075f07633cad402a5b6"), "marca" : "nisan", "modelo" : "note", "color" : "gris", "dimensiones" : { "altuta" : 1.6, "ancho" : 1.78, "largo" : 3 }, "matricula" : "2942PLA" }
{ "_id" : ObjectId("6162c075f07633cad402a5b7"), "marca" : "volkswagen", "modelo" : "tiguan", "color" : "negro", "dimensiones" : { "altuta" : 1.6, "ancho" : 1.77, "largo" : 3.9 }, "matricula" : "5637JNE" }
{ "_id" : ObjectId("6162c075f07633cad402a5b8"), "marca" : "maserati", "modelo" : "grecale", "color" : "verde", "dimensiones" : { "altuta" : 1.3, "ancho" : 1.72, "largo" : 4.3 }, "matricula" : "1278SKR" }
{ "_id" : ObjectId("6162c075f07633cad402a5b9"), "marca" : "opel", "modelo" : "corsa", "color" : "rojo", "dimensiones" : { "altuta" : 1, "ancho" : 0.83, "largo" : 2.2 }, "matricula" : "0364FJE" }
>
```

Como vemos la información se ha cargado correctamente, ya tenemos nuestra base de datos con datos en las dos colecciones.

3.2.Consultas en la base de datos.

Ahora que tenemos nuestra base de datos con sus respectivos documentos en sus respectivas colecciones, vamos a realizar consultas sobre ella usando algunos comando y operadores query.

Para ello vamos a crear un archivo json llamado consultas, en el que meteremos todas las consultas con comentarios diciendo lo que hace cada una, para realizar la consulta copiamos el comando y lo pegamos en la power shell.

Ya hemos visto con el ejemplo anterior el comando find, que es el que usaremos para hacer las consultas. Vamos a realizar una consulta que nos muestre todos los coches mercedes, para ello ejecutamos lo siguiente:

```
> db.coches.find( { marca: "mercedes" } )
{ "_id" : ObjectId("6162c075f07633cad402a5b4"), "marca" : "mercedes", "modelo" : "amg", "color" : "negro", "dimensiones" : { "altuta" : 1.5, "ancho" : 1.85, "largo" : 4.3 }, "matricula" : "4937KHF" }
>
```

Ahora añadimos esta consulta a nuestro documento.


```

1  /*
2  Mostrar todos los coches de la marca mercedes.
3  */
4  db.coches.find( { marca: "mercedes" } )
5

```

Ahora vamos a hacer una consulta que muestre los coches de la marca mercedes o chevrolet, para ello usaremos el operador \$in.

```

> db.coches.find( { marca: { $in: [ "mercedes", "chevrolet" ] } } )
{ "_id" : ObjectId("6162c075f07633cad402a5b0"), "marca" : "chevrolet", "modelo" : "cruze", "color" : "gris",
  "dimensiones" : { "altuta" : 1.4, "ancho" : 1.9, "largo" : 4.3 }, "matricula" : "8749JEH" }
{ "_id" : ObjectId("6162c075f07633cad402a5b4"), "marca" : "mercedes", "modelo" : "amg", "color" : "negro",
  "dimensiones" : { "altuta" : 1.5, "ancho" : 1.85, "largo" : 4.3 }, "matricula" : "4937KHF" }
>

```

El operador in se usa para buscar varios valores de un mismo campo mientras que para hacer esta consulta con varios campos distintos usamos el operador \$or. Realizamos una consulta que nos muestre todas las motos de color negro o de 1m de altura.

```

> db.motos.find( { $or: [ { color: "negro" }, { "dimensiones.altuta": 1 } ] } )
{ "_id" : ObjectId("6161b82843692ee748226c99"), "marca" : "ducati", "modelo" : "scrambler", "color" : "rojo",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "1764HFK" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a6"), "marca" : "kawasaki", "modelo" : "eliminator", "color" : "negro",
  "dimensiones" : { "altuta" : 1.3, "ancho" : 0.9, "largo" : 2.3 }, "matricula" : "37655FW" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a7"), "marca" : "Harley", "modelo" : "883", "color" : "rojo",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "9832FHE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a9"), "marca" : "Suzuki", "modelo" : "rouser", "color" : "negro",
  "dimensiones" : { "altuta" : 0.9, "ancho" : 0.79, "largo" : 2 }, "matricula" : "9873EHJ" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ab"), "marca" : "kawasaki", "modelo" : "mach 3", "color" : "blanco",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.8, "largo" : 2 }, "matricula" : "9837EJD" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ac"), "marca" : "ducati", "modelo" : "scrambler", "color" : "gris",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "2982PLA" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ad"), "marca" : "Harley", "modelo" : "sportster", "color" : "negro",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.77, "largo" : 1.9 }, "matricula" : "9843JNE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5af"), "marca" : "Suzuki", "modelo" : "santana", "color" : "rojo",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.83, "largo" : 2.2 }, "matricula" : "9275FJE" }
>

```

Ahora realizamos una consulta que nos muestre todas las motos que no son negras, para ello usamos el operador \$ne.

```

> db.motos.find({color: {$ne: "negro" } } )
{ "_id" : ObjectId("6161b82843692ee748226c99"), "marca" : "ducati", "modelo" : "scrambler", "color" : "rojo",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "1764HFK" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a7"), "marca" : "Harley", "modelo" : "883", "color" : "rojo",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "9832FHE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a8"), "marca" : "Harley", "modelo" : "FXDR 114", "color" : "blanco",
  "dimensiones" : { "altuta" : 1.4, "ancho" : 1, "largo" : 2.2 }, "matricula" : "8675RNS" }
{ "_id" : ObjectId("6162bc51f07633cad402a5aa"), "marca" : "ducati", "modelo" : "diana", "color" : "rojo",
  "dimensiones" : { "altuta" : 0.9, "ancho" : 0.75, "largo" : 1.8 }, "matricula" : "8904LAI" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ab"), "marca" : "kawasaki", "modelo" : "mach 3", "color" : "blanco",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.8, "largo" : 2 }, "matricula" : "9837EJD" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ac"), "marca" : "ducati", "modelo" : "scrambler", "color" : "gris",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "2982PLA" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ae"), "marca" : "kawasaki", "modelo" : "ER6", "color" : "verde",
  "dimensiones" : { "altuta" : 0.9, "ancho" : 0.72, "largo" : 1.8 }, "matricula" : "1278DHE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5af"), "marca" : "Suzuki", "modelo" : "santana", "color" : "rojo",
  "dimensiones" : { "altuta" : 1, "ancho" : 0.83, "largo" : 2.2 }, "matricula" : "9275FJE" }
>

```

Consulta que muestra todos los coches con altura mayor a 1,4m con el operador \$gt.

```
> db.coches.find( { "dimensiones.altuta": { $gt: 1.4 } } )
{ "_id" : ObjectId("6161b94e43692ee748226c9a"), "marca" : "opel", "modelo" : "corsa", "color" : "gris", "dimensiones" : { "altuta" : 1.45, "ancho" : 1.78, "largo" : 3.5 }, "matricula" : "1890SDF" }
{ "_id" : ObjectId("6162c075f07633cad402a5b1"), "marca" : "mazda", "modelo" : "10", "color" : "rojo", "dimensiones" : { "altuta" : 1.5, "ancho" : 1.78, "largo" : 4 }, "matricula" : "9083JEA" }
{ "_id" : ObjectId("6162c075f07633cad402a5b4"), "marca" : "mercedes", "modelo" : "amg", "color" : "negro", "dimensiones" : { "altuta" : 1.5, "ancho" : 1.85, "largo" : 4.3 }, "matricula" : "4937KHF" }
{ "_id" : ObjectId("6162c075f07633cad402a5b6"), "marca" : "nisan", "modelo" : "note", "color" : "gris", "dimensiones" : { "altuta" : 1.6, "ancho" : 1.78, "largo" : 3 }, "matricula" : "2942PLA" }
{ "_id" : ObjectId("6162c075f07633cad402a5b7"), "marca" : "wolkswagen", "modelo" : "tiguan", "color" : "negro", "dimensiones" : { "altuta" : 1.6, "ancho" : 1.77, "largo" : 3.9 }, "matricula" : "5637JNE" }
>
```

También se puede hacer esta consulta incluyendo a los coches que tengan 1.4m con el operador \$gte.

```
> db.coches.find( { "dimensiones.altuta": { $gte: 1.4 } } )
{ "_id" : ObjectId("6161b94e43692ee748226c9a"), "marca" : "opel", "modelo" : "corsa", "color" : "gris", "dimensiones" : { "altuta" : 1.45, "ancho" : 1.78, "largo" : 3.5 }, "matricula" : "1890SDF" }
{ "_id" : ObjectId("6162c075f07633cad402a5b0"), "marca" : "chevrolet", "modelo" : "cruze", "color" : "gris", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.9, "largo" : 4.3 }, "matricula" : "8749JEH" }
{ "_id" : ObjectId("6162c075f07633cad402a5b1"), "marca" : "mazda", "modelo" : "10", "color" : "rojo", "dimensiones" : { "altuta" : 1.5, "ancho" : 1.78, "largo" : 4 }, "matricula" : "9083JEA" }
{ "_id" : ObjectId("6162c075f07633cad402a5b2"), "marca" : "audi", "modelo" : "q3", "color" : "blanco", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.96, "largo" : 3.9 }, "matricula" : "3576AEG" }
{ "_id" : ObjectId("6162c075f07633cad402a5b3"), "marca" : "bmw", "modelo" : "m2", "color" : "azul", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.79, "largo" : 4.2 }, "matricula" : "6287AHE" }
{ "_id" : ObjectId("6162c075f07633cad402a5b4"), "marca" : "mercedes", "modelo" : "amg", "color" : "negro", "dimensiones" : { "altuta" : 1.5, "ancho" : 1.85, "largo" : 4.3 }, "matricula" : "4937KHF" }
{ "_id" : ObjectId("6162c075f07633cad402a5b5"), "marca" : "seat", "modelo" : "ibiza", "color" : "blanco", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.8, "largo" : 3.2 }, "matricula" : "9836HJR" }
{ "_id" : ObjectId("6162c075f07633cad402a5b6"), "marca" : "nisan", "modelo" : "note", "color" : "gris", "dimensiones" : { "altuta" : 1.6, "ancho" : 1.78, "largo" : 3 }, "matricula" : "2942PLA" }
{ "_id" : ObjectId("6162c075f07633cad402a5b7"), "marca" : "wolkswagen", "modelo" : "tiguan", "color" : "negro", "dimensiones" : { "altuta" : 1.6, "ancho" : 1.77, "largo" : 3.9 }, "matricula" : "5637JNE" }
>
```

Podemos hacer una consulta con valores menores al especificado con \$lt, en esta consulta veremos las motos con un ancho menor de 0.8m.

```
> db.motos.find( { "dimensiones.ancho": { $lt: 0.8 } } )
{ "_id" : ObjectId("6161b82843692ee748226c99"), "marca" : "ducati", "modelo" : "scrambler", "color" : "rojo", "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "1764HFK" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a7"), "marca" : "Harley", "modelo" : "883", "color" : "rojo", "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "9832FHE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a9"), "marca" : "Suzuki", "modelo" : "rouser", "color" : "negro", "dimensiones" : { "altuta" : 0.9, "ancho" : 0.79, "largo" : 2 }, "matricula" : "9873EHJ" }
{ "_id" : ObjectId("6162bc51f07633cad402a5aa"), "marca" : "ducati", "modelo" : "diana", "color" : "rojo", "dimensiones" : { "altuta" : 0.9, "ancho" : 0.75, "largo" : 1.8 }, "matricula" : "8904LAI" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ac"), "marca" : "ducati", "modelo" : "scrambler", "color" : "gris", "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "2982PLA" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ad"), "marca" : "Harley", "modelo" : "sportster", "color" : "negro", "dimensiones" : { "altuta" : 1, "ancho" : 0.77, "largo" : 1.9 }, "matricula" : "9843JNE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ae"), "marca" : "kawasaki", "modelo" : "ER6", "color" : "verde", "dimensiones" : { "altuta" : 0.9, "ancho" : 0.72, "largo" : 1.8 }, "matricula" : "1278DHE" }
>
```

Al igual que con el operador \$gt también podemos incluir el valor con \$lte

```
> db.motos.find( { "dimensiones.anchos": { $lte: 0.8 } } )
{ "_id" : ObjectId("6161b82843692ee748226c99"), "marca" : "ducati", "modelo" : "scrambler", "color" : "rojo", "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "1764HFK" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a7"), "marca" : "Harley", "modelo" : "883", "color" : "rojo", "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "9832FHE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5a9"), "marca" : "Suzuki", "modelo" : "rouser", "color" : "negro", "dimensiones" : { "altuta" : 0.9, "ancho" : 0.79, "largo" : 2 }, "matricula" : "9873EHJ" }
{ "_id" : ObjectId("6162bc51f07633cad402a5aa"), "marca" : "ducati", "modelo" : "diana", "color" : "rojo", "dimensiones" : { "altuta" : 0.9, "ancho" : 0.75, "largo" : 1.8 }, "matricula" : "8904LAI" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ab"), "marca" : "kawasaki", "modelo" : "mach 3", "color" : "blanco", "dimensiones" : { "altuta" : 1, "ancho" : 0.8, "largo" : 2 }, "matricula" : "9837EJD" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ac"), "marca" : "ducati", "modelo" : "scrambler", "color" : "gris", "dimensiones" : { "altuta" : 1, "ancho" : 0.78, "largo" : 2 }, "matricula" : "2982PLA" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ad"), "marca" : "Harley", "modelo" : "sportster", "color" : "negro", "dimensiones" : { "altuta" : 1, "ancho" : 0.77, "largo" : 1.9 }, "matricula" : "9843JNE" }
{ "_id" : ObjectId("6162bc51f07633cad402a5ae"), "marca" : "kawasaki", "modelo" : "ER6", "color" : "verde", "dimensiones" : { "altuta" : 0.9, "ancho" : 0.72, "largo" : 1.8 }, "matricula" : "1278DHE" }
>
```

Por último podemos buscar documentos que no tengan los valores especificados con el operador \$nin , vamos a realizar una consulta de los coches que no sean mercedes.

```
> db.coches.find( { marca:{ $nin: [ "mercedes", "chevrolet" ] } } )
{ "_id" : ObjectId("6161b94e43692ee748226c9a"), "marca" : "opel", "modelo" : "corsa", "color" : "gris", "dimensiones" : { "altuta" : 1.45, "ancho" : 1.78, "largo" : 3.5 }, "matricula" : "1890SDF" }
{ "_id" : ObjectId("6162c075f07633cad402a5b1"), "marca" : "mazda", "modelo" : "10", "color" : "rojo", "dimensiones" : { "altuta" : 1.5, "ancho" : 1.78, "largo" : 4 }, "matricula" : "9083JEA" }
{ "_id" : ObjectId("6162c075f07633cad402a5b2"), "marca" : "audi", "modelo" : "q3", "color" : "blanco", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.96, "largo" : 3.9 }, "matricula" : "3576AEG" }
{ "_id" : ObjectId("6162c075f07633cad402a5b3"), "marca" : "bmw", "modelo" : "m2", "color" : "azul", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.79, "largo" : 4.2 }, "matricula" : "6287AHE" }
{ "_id" : ObjectId("6162c075f07633cad402a5b5"), "marca" : "seat", "modelo" : "ibiza", "color" : "blanco", "dimensiones" : { "altuta" : 1.4, "ancho" : 1.8, "largo" : 3.2 }, "matricula" : "9836HJR" }
{ "_id" : ObjectId("6162c075f07633cad402a5b6"), "marca" : "nisan", "modelo" : "note", "color" : "gris", "dimensiones" : { "altuta" : 1.6, "ancho" : 1.78, "largo" : 3 }, "matricula" : "2942PLA" }
{ "_id" : ObjectId("6162c075f07633cad402a5b7"), "marca" : "volkswagen", "modelo" : "tiguan", "color" : "negro", "dimensiones" : { "altuta" : 1.6, "ancho" : 1.77, "largo" : 3.9 }, "matricula" : "5637JNE" }
{ "_id" : ObjectId("6162c075f07633cad402a5b8"), "marca" : "maserati", "modelo" : "grecale", "color" : "verde", "dimensiones" : { "altuta" : 1.3, "ancho" : 1.72, "largo" : 4.3 }, "matricula" : "1278SKR" }
{ "_id" : ObjectId("6162c075f07633cad402a5b9"), "marca" : "opel", "modelo" : "corsa", "color" : "rojo", "dimensiones" : { "altuta" : 1, "ancho" : 0.83, "largo" : 2.2 }, "matricula" : "0364FJE" }
>
```

El archivo quedaría así:

```
1 /*
2 Mostrar todos los coches de la marca mercedes.
3 */
4 db.coches.find( { marca: "mercedes" } )
5
6 /*
7 Mostrar todos los coches de la marca mercedes y chevrolet.
8 */
9 db.coches.find( { marca:{ $nin: [ "mercedes", "chevrolet" ] } } )
10
11 /*
12 Mostrar todas las motos de color negro o de 1 metro de alto.
13 */
14 db.motos.find( { $or: [ { color: "negro" }, { "dimensiones.altuta": 1 } ] } )
15
16 /*
17 Mostrar todos los coches con altura mayor a 1.4 metros.
18 */
19 db.coches.find( { "dimensiones.altuta": { $gt: 1.4 } } )
20
21 /*
22 Mostrar todas las motos con un ancho menor de 0.8.
23 */
24 db.motos.find( { "dimensiones.anchos": { $lt: 0.8 } } )
25
26 /*
27 Mostrar todos los coches que no sean mercedes ni chevrolet.
28 */
29 db.coches.find( { marca:{ $nin: [ "mercedes", "chevrolet" ] } } )
30
```