

Primer proyecto Visual Studio Code y MongoDB

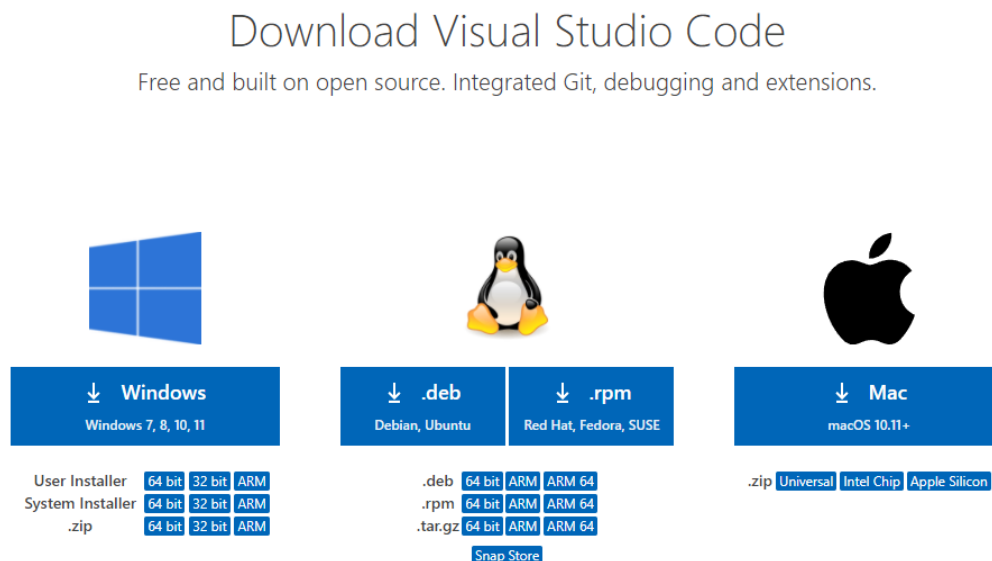


Índice

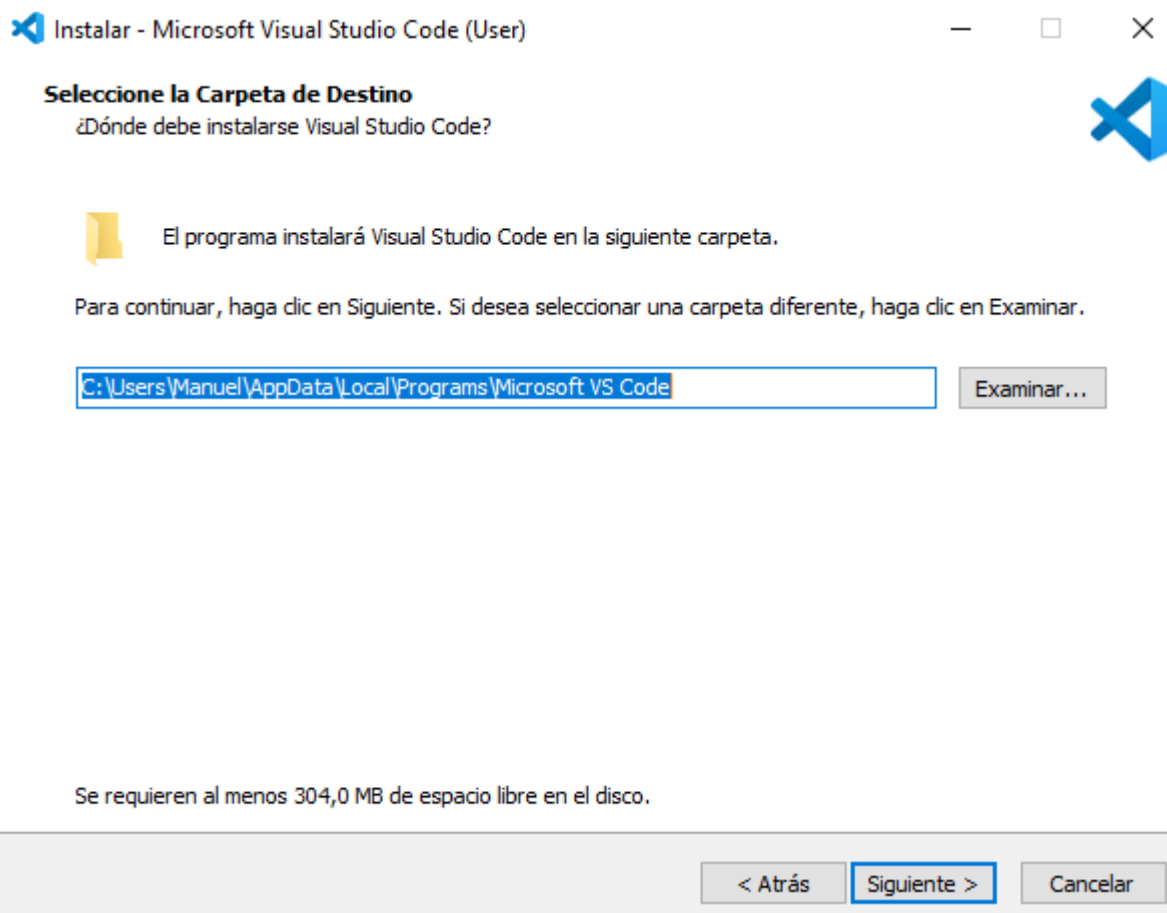
Instalación de Visual Studio Code	3
Instalación de MongoDB	5
Comandos en Mongo Shell	12
Métodos y operadores	14

Instalación de Visual Studio Code

Para instalar Visual Studio Code nos dirigimos a <https://code.visualstudio.com/Download> y descargamos el archivo de instalación.



Después ejecutamos el archivo de instalación una vez que se haya descargado, aceptaremos el acuerdo de licencia y continuamos, luego elegimos la dirección donde se guardará el programa.



A continuación seleccionamos la carpeta del Menú de Inicio y seleccionamos las tareas adicionales que deseemos, podemos crear un acceso directo en el escritorio. Por último haremos click en instalar y comenzará la instalación. Habrá que darle a finalizar para acabar.

Instalar - Microsoft Visual Studio Code (User) — □ ×

Completando la instalación de Visual Studio Code

El programa completó la instalación de Visual Studio Code en su sistema. Puede ejecutar la aplicación utilizando los accesos directos creados.

Haga clic en Finalizar para salir del programa de instalación.

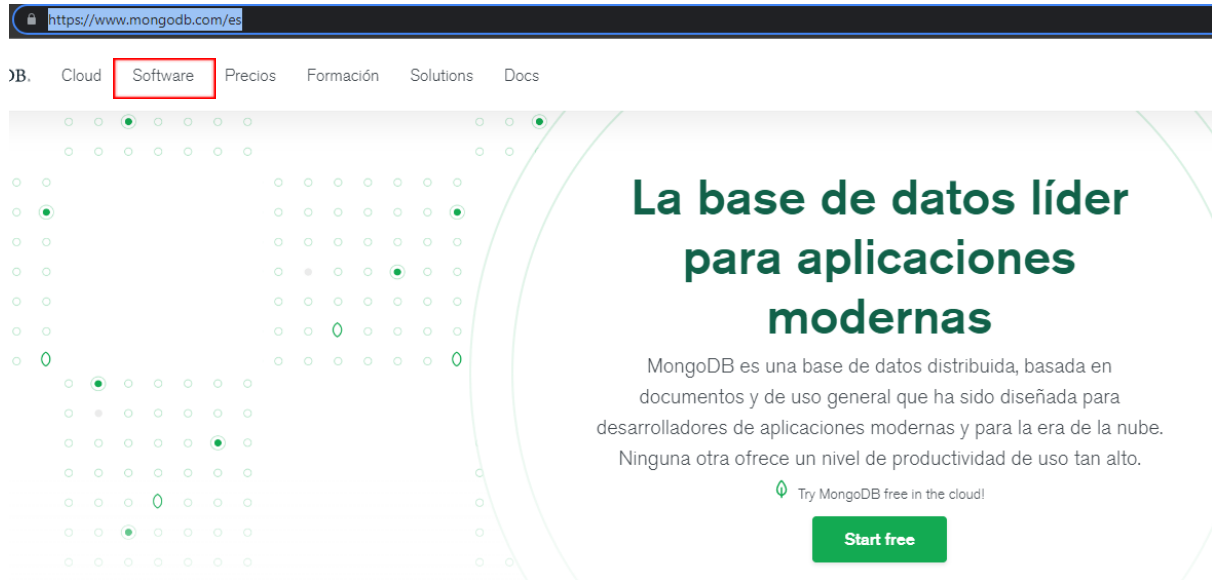


☒ Ejecutar Visual Studio Code

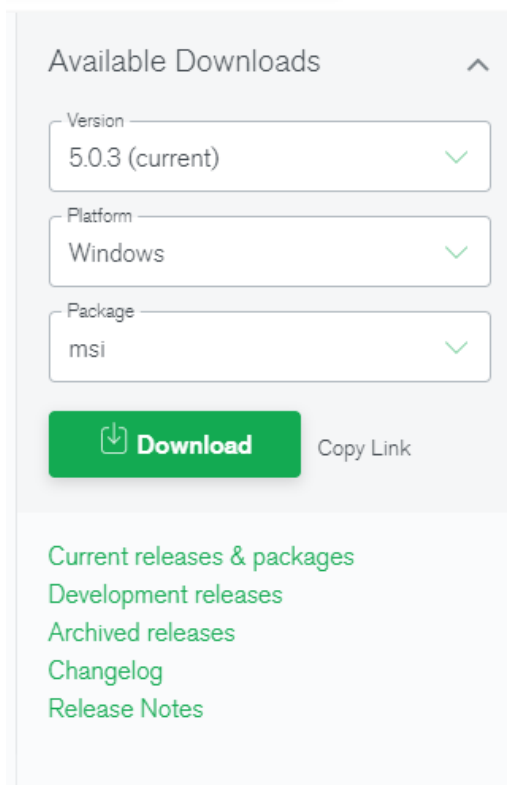
Finalizar

Instalación de MongoDB

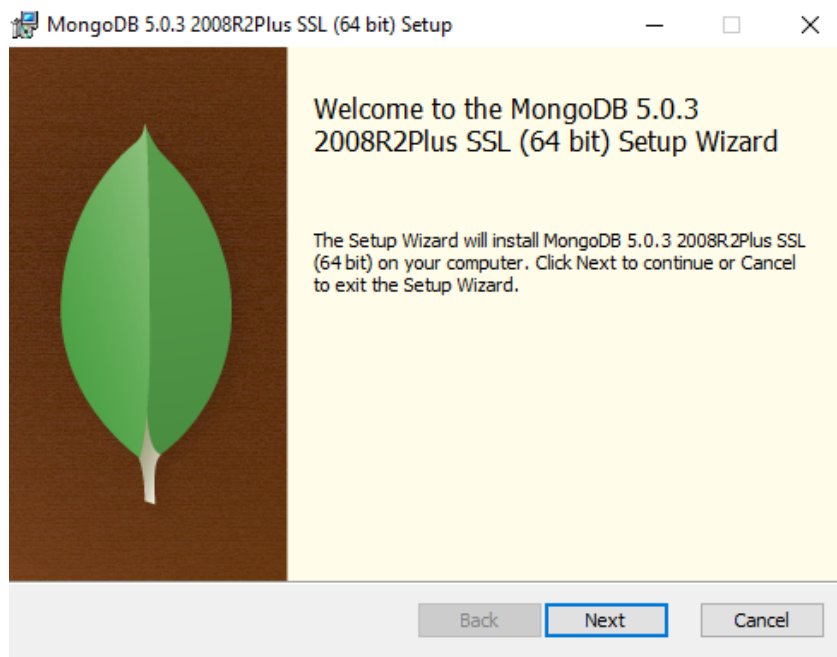
Primero iremos a la página oficial de MongoDB, <https://www.mongodb.com/es> y nos dirigiremos a Software>Community Server.



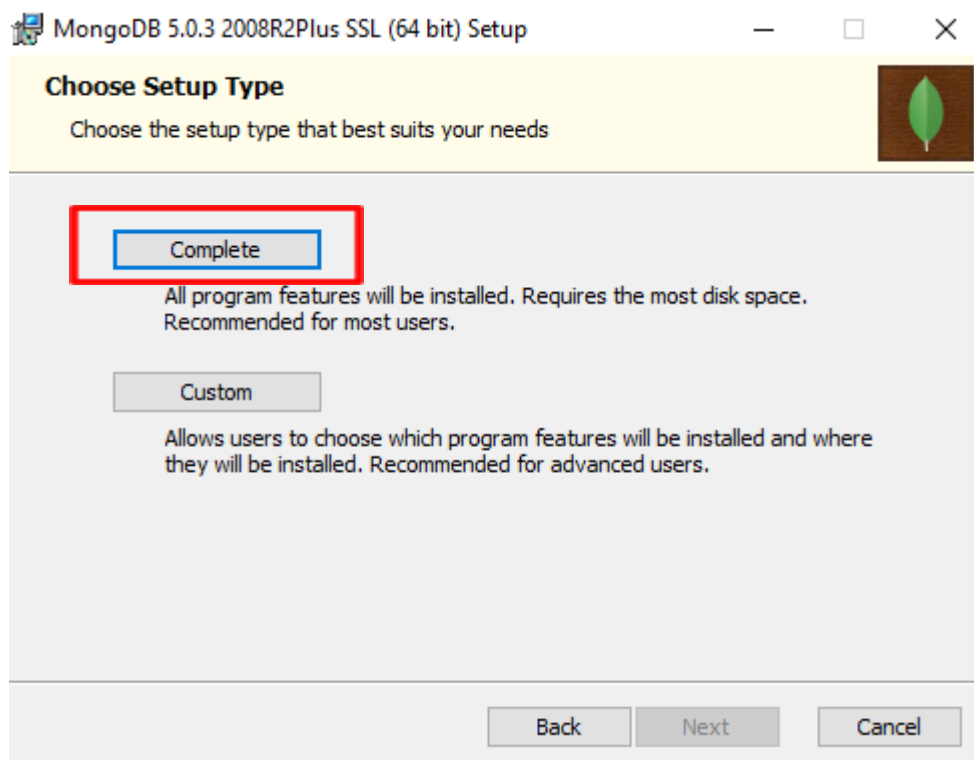
Descargamos la versión más reciente



Empezamos la instalación:



Aceptamos los términos de licencia y continuamos. Elegimos la instalación completa



Instalamos MongoDB como un servicio

MongoDB 5.0.3 2008R2Plus SSL (64 bit) Service Customization

Service Configuration
Specify optional settings to configure MongoDB as a service.

☒ Install MongoDB as a Service

☒ Run service as Network Service user

☐ Run service as a local or domain user:

Account Domain:

Account Name:

Account Password:

Service Name:

Data Directory:

Log Directory:

< Back Next > Cancel

Instalamos Mongo Compass

MongoDB Compass

Install MongoDB Compass

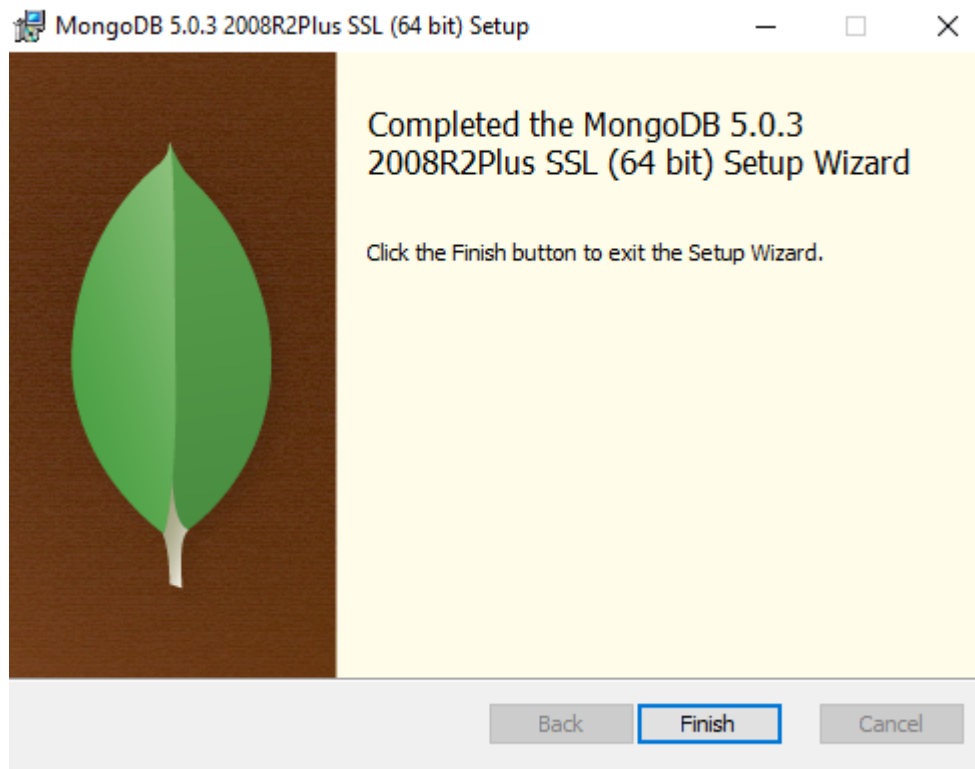
MongoDB Compass is the official graphical user interface for MongoDB.

By checking below this installer will automatically download and install the latest version of MongoDB Compass on this machine. You can learn more about MongoDB Compass here: <https://www.mongodb.com/products/comp...>

☒ Install MongoDB Compass

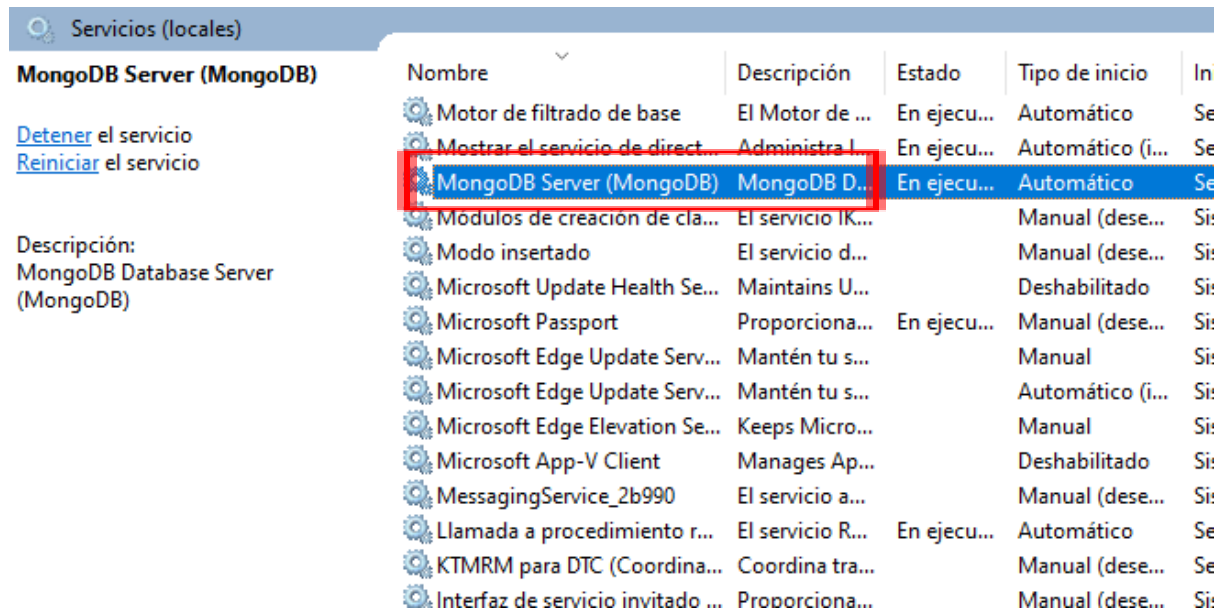
Back Next Cancel

Lo instalamos y finalizamos la instalación

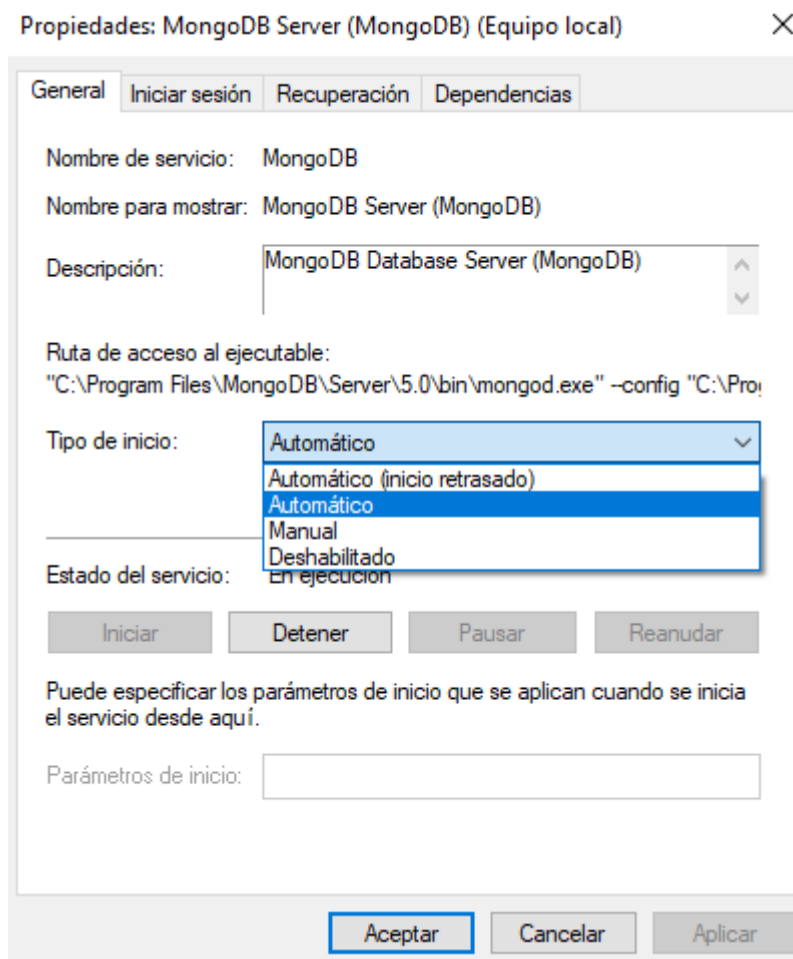


Después de la instalación nos dirigimos a los servicios de Windows y configuramos el servicio de MongoDB. Para ello buscamos “Servicios” en el buscador de Windows.

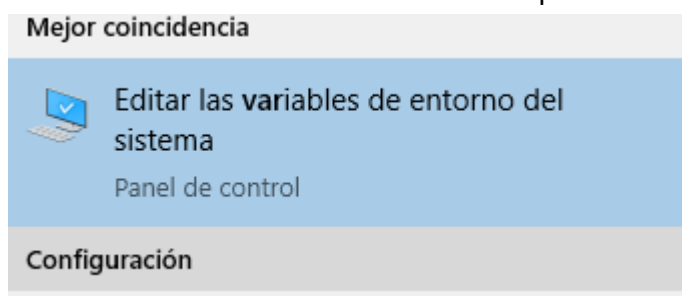
Hacemos clic en cualquier servicio y escribimos “m” para buscar el servicio Mongo

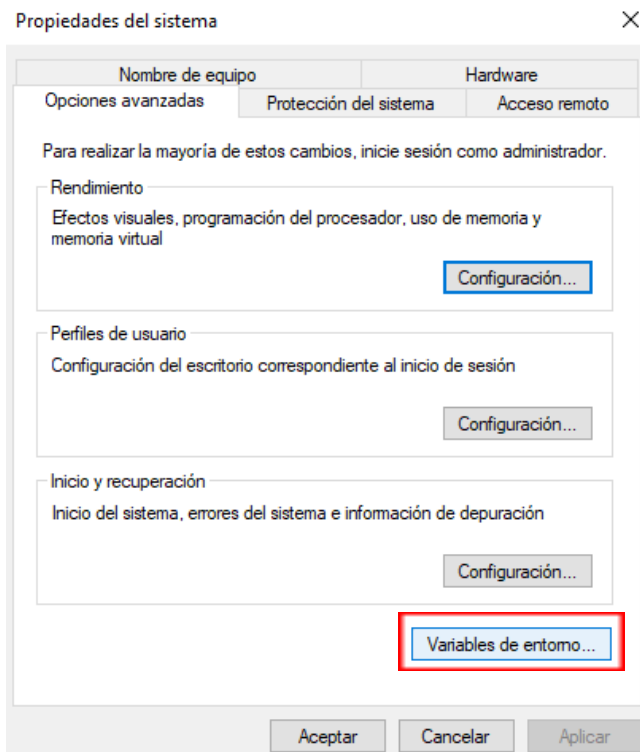


Comprobamos que esté en automático para no tener problemas cuando usemos MongoDB

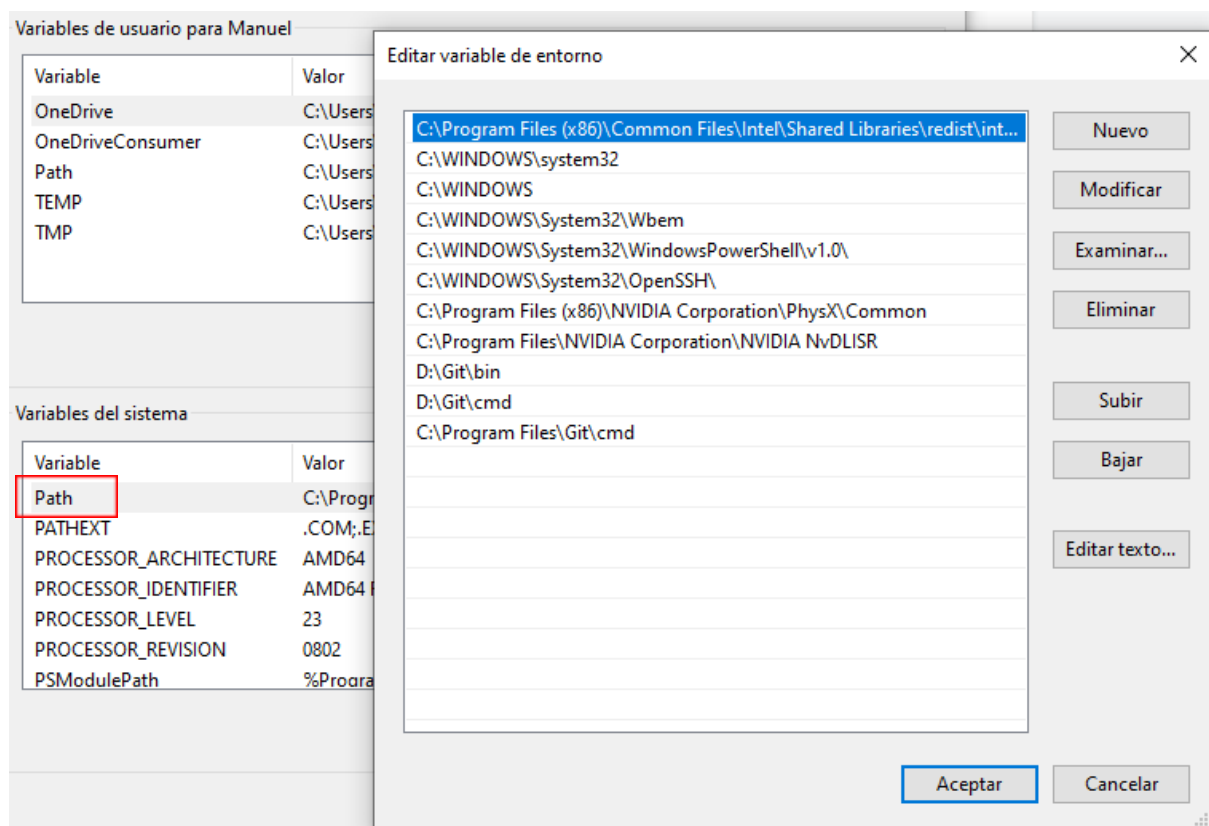


Después de configurar los servicios, debemos ir a las variables de entorno del sistema y añadir una nueva entrada en la variable “path”

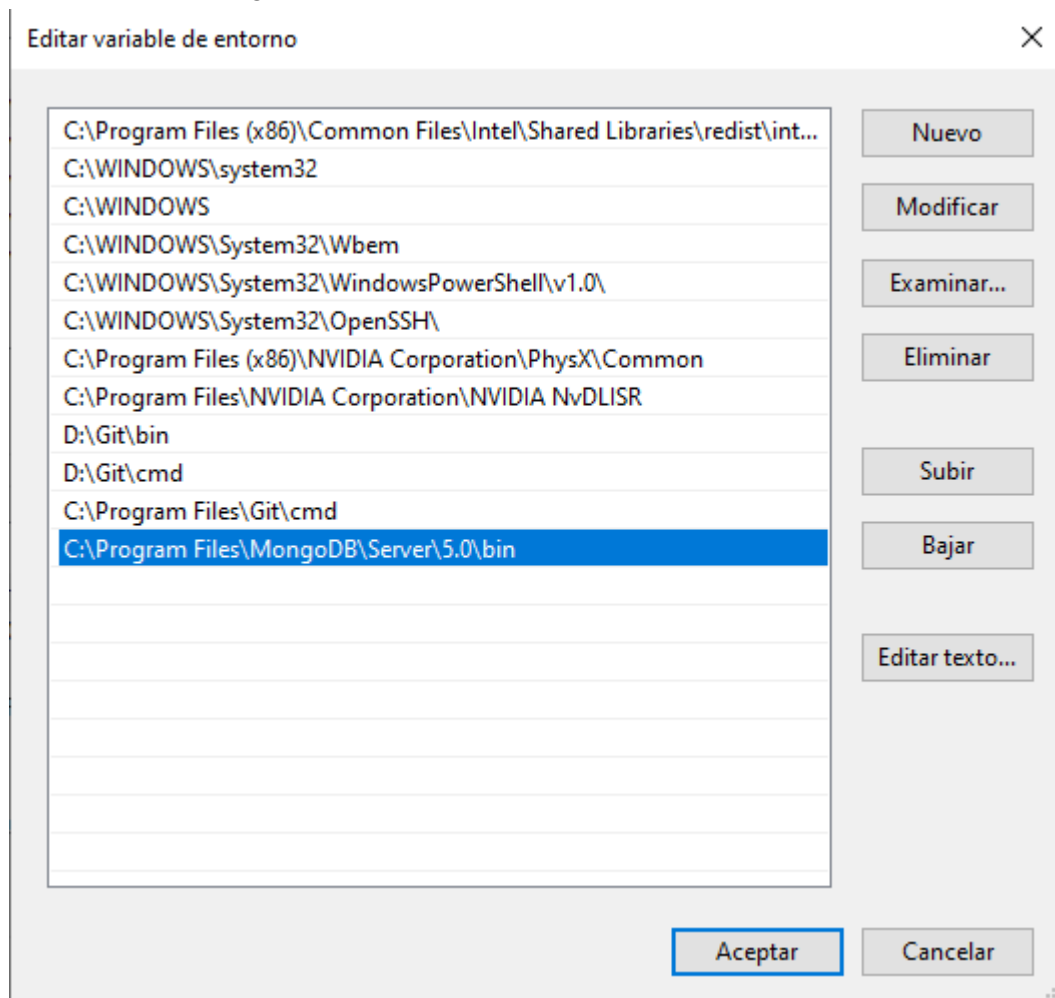




Hacemos doble click en la variable del sistema "Path"



Le daremos a nuevo y ponemos la ubicación en la que se encuentra la carpeta bin de la instalación de MongoDB.



Ya podemos usar MongoDB Compass y podemos comprobar la versión de Mongo en la línea de comandos mediante el comando “mongo --version”.

Si vamos a usar Mongodb Shell tenemos que descargarlo en el siguiente enlace: <https://www.mongodb.com/products/shell>, y ejecutar el archivo de instalación. Después deberás configurar los servicios y las variables de entorno de MongoDB Shell como hicimos anteriormente, esta vez usando la ubicación donde hayas instalado MongoDB Shell.

Comandos en Mongo Shell

En Windows PowerShell nos ubicamos en la carpeta del proyecto donde estamos diseñando la colección mediante el comando “cd”

```
PS C:\Users\Manuel\Desktop\ASIR1\GBD\VSCmongoDB_01> cd C:\Users\Manuel\Desktop\ASIR1\GBD\VSCmongoDB_01\src
PS C:\Users\Manuel\Desktop\ASIR1\GBD\VSCmongoDB_01\src>
```

Si escribimos “dir” vemos el contenido de la carpeta “src”

```
PS C:\Users\Manuel\Desktop\ASIR1\GBD\VSCmongoDB_01\src> dir

Directorio: C:\Users\Manuel\Desktop\ASIR1\GBD\VSCmongoDB_01\src

Mode                LastWriteTime         Length Name
----                -
-a----             17/10/2021   20:07           363 consultas.js
-a----             17/10/2021   20:24           752 insert.js

PS C:\Users\Manuel\Desktop\ASIR1\GBD\VSCmongoDB_01\src>
```

Después ejecutamos Mongo Shell escribiendo “mongosh”

```
PS C:\Users\Manuel\Desktop\ASIR1\GBD\VSCmongoDB_01\src> mongosh
Current Mongosh Log ID: 616c70d6cd970c49b7b54154
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Using MongoDB:      5.0.3
Using Mongosh:       1.1.0

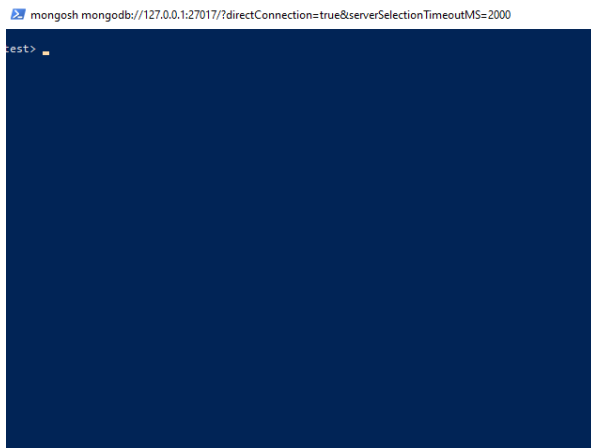
For mongosh info see: https://docs.mongodb.com/mongosh-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting:
  2021-10-17T20:37:52.296+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
test>
```

Ahora podemos escribir “cls” para limpiar y vaciar las líneas de comando.



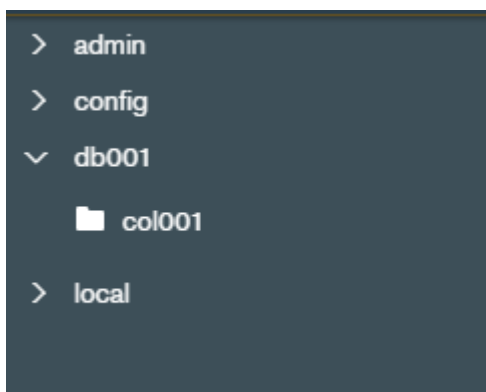
Mediante el comando `show dbs` podemos ver las bases de datos que hay creadas. Podemos crear una mediante el comando `use` y el nombre que le queremos dar a la base de datos, en mi caso sería `use db001`.

Mediante el comando `db.nombredecoleccion.insertOne()` podemos crear una colección

Para subir la colección que hemos diseñado en Visual Studio Code escribimos el comando: `load ("insert.js")`

```
test> use db001
switched to db db001
db001> load ("insert.js")
true
db001>
```

Ahora abriremos MongoDB Compass y veremos que se ha creado la base de datos "db001", y dentro de esta base de datos está la colección 001



Métodos y operadores

Hay varios métodos que se pueden utilizar para crear documentos y para eliminarlos. Incluso pueden crear una colección si esta aún no existe.

- El método **insertOne**: `db.coleccion.insertOne()` inserta un solo documento dentro de una colección.

```
db.col001.insertOne({Trabajadores: 7})
```

- El método **insertMany**: `db.coleccion.insertMany()` inserta varios documentos a la vez en una colección y pasa una serie de documentos al método.

```
db.col001.insertMany([
  {name: "Elena", country: "Nicaragua", age: 55, size: {h: 1.75, w: 80}, shift: "morning"},
  {name: "Pablo", country: "Spain", age: 48, size: {h: 1.83, w: 72}, shift: "morning"},
  {name: "Manuel", country: "Spain", age: 29, size: {h: 1.94, w: 94}, shift: "morning"},
  {name: "Mohammed", country: "Morocco", age: 30, size: {h: 1.92, w: 78}, shift: "morning"},
  {name: "Juana", country: "Peru", age: 35, size: {h: 1.60, w: 70}, shift: "afternoon"},
  {name: "Carmen", country: "Spain", age: 48, size: {h: 1.73, w: 60}, shift: "afternoon"},
  {name: "Pablo", country: "Colombia", age: 43, size: {h: 1.78, w: 75}, shift: "afternoon"},
])
```

Hay que escribir los documentos dentro del array, es decir entre corchetes (`[{documentos}]`)

También existen métodos para hacer consultas y filtrar los documentos. Para hacer esto se usa el método **find**:

- Si queremos buscar todos los documentos con todos los campos, sin filtrar nada, se usa el método **db.coleccion.find()**

```

db001> db.col001.find()
[
  {
    _id: ObjectId("616c76c3358f97378ffab7fc"),
    name: 'Elena',
    country: 'Nicaragua',
    age: 55,
    size: { h: 1.75, w: 80 },
    shift: 'morning'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab7fd"),
    name: 'Pablo',
    country: 'Spain',
    age: 48,
    size: { h: 1.83, w: 72 },
    shift: 'morning'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab7fe"),
    name: 'Manuel',
    country: 'Spain',
    age: 29,
    size: { h: 1.94, w: 94 },
    shift: 'morning'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab7ff"),
    name: 'Mohammed',
    country: 'Morocco',
    age: 30,
    size: { h: 1.92, w: 78 },
    shift: 'morning'
  },
]

```

- Si queremos buscar documentos con un filtro debemos usar el método `db.coleccion.find({campo: "valor"})`. Por ejemplo:

```

db001> db.col001.find({shift: "afternoon"})
[
  {
    _id: ObjectId("616c76c3358f97378ffab800"),
    name: 'Juana',
    country: 'Peru',
    age: 35,
    size: { h: 1.6, w: 70 },
    shift: 'afternoon'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab801"),
    name: 'Carmen',
    country: 'Spain',
    age: 48,
    size: { h: 1.73, w: 60 },
    shift: 'afternoon'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab802"),
    name: 'Pablo',
    country: 'Colombia',
    age: 43,
    size: { h: 1.78, w: 75 },
    shift: 'afternoon'
  }
]

```

También se puede hacer esto usando el operador **\$eq**:

```

db001> db.col001.find({shift: {$eq: "afternoon"}})
[
  {
    _id: ObjectId("616c76c3358f97378ffab800"),
    name: 'Juana',
    country: 'Peru',
    age: 35,
    size: { h: 1.6, w: 70 },
    shift: 'afternoon'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab801"),
    name: 'Carmen',
    country: 'Spain',
    age: 48,
    size: { h: 1.73, w: 60 },
    shift: 'afternoon'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab802"),
    name: 'Pablo',
    country: 'Colombia',
    age: 43,
    size: { h: 1.78, w: 75 },
    shift: 'afternoon'
  }
]

```

El operador **\$eq** se puede ignorar ya que no hace falta usarlo para filtrar documentos, se puede hacer como en el ejemplo 1.

Si queremos buscar un documento que cumpla con un campo que está dentro de otro campo debemos usar la siguiente sintaxis: `db.coleccion.find({campo1: {campo2: valor, campo3: valor2}, })`


```
db001> db.col001.find({size: {h: 1.60, w: 70}, })
[
  {
    _id: ObjectId("616c76c3358f97378ffab800"),
    name: 'Juana',
    country: 'Peru',
    age: 35,
    size: { h: 1.6, w: 70 },
    shift: 'afternoon'
  }
]
db001>
```

- Si queremos buscar un documento que cumpla con un valor mayor o igual de cierto valor se usa el operador **\$gte**:

```
db001> db.col001.find({age: {$gte: 35}})
[
  {
    _id: ObjectId("616c76c3358f97378ffab7fc"),
    name: 'Elena',
    country: 'Nicaragua',
    age: 55,
    size: { h: 1.75, w: 80 },
    shift: 'morning'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab7fd"),
    name: 'Pablo',
    country: 'Spain',
    age: 48,
    size: { h: 1.83, w: 72 },
    shift: 'morning'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab800"),
    name: 'Juana',
    country: 'Peru',
    age: 35,
    size: { h: 1.6, w: 70 },
    shift: 'afternoon'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab801"),
    name: 'Larmen',
    country: 'Spain',
    age: 48,
    size: { h: 1.73, w: 60 },
    shift: 'afternoon'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab802"),
    name: 'Pablo',
    country: 'Colombia',
    age: 43,
    size: { h: 1.78, w: 75 },
    shift: 'afternoon'
  }
]
```

Si queremos que busque un documento que sea mayor que el valor que queramos se usa el operador **\$gt**

- Si queremos buscar un documento que cumpla con un valor menor o igual que un valor específico se usa el operador **\$lte**:

```
db001> db.col001.find({age: {$lte: 35}})
[
  {
    _id: ObjectId("616c76c3358f97378ffab7fe"),
    name: 'Manuel',
    country: 'Spain',
    age: 29,
    size: { h: 1.94, w: 94 },
    shift: 'morning'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab7ff"),
    name: 'Mohammed',
    country: 'Morocco',
    age: 30,
    size: { h: 1.92, w: 78 },
    shift: 'morning'
  },
  {
    _id: ObjectId("616c76c3358f97378ffab800"),
    name: 'Juana',
    country: 'Peru',
    age: 35,
    size: { h: 1.6, w: 70 },
    shift: 'afternoon'
  }
]
```

Si queremos que cumpla con un valor menor que otro se usa el operador **\$lt**.