

# Primeros pasos MongoDB



Índice:

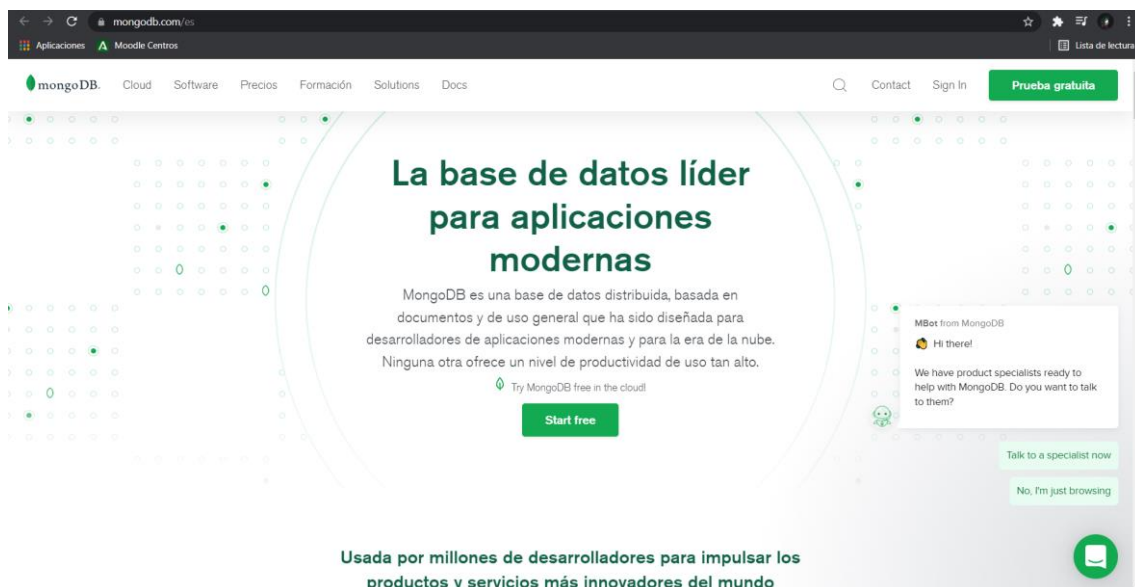
- 1. ¿Qué es MongoDB? ..... 2
- 2. Instalacion Mongo:..... 2
- 3. Instalación Visual Studio Code. .... 10
  - 3.1 Creación de un proyecto desde Visual Studio Code. .... 13
  - 3.2 Antes de comenzar ..... 15
- 4. Insert, Query, Update y Delete. .... 19
  - 4.1 Insert Documents. .... 19
  - 4.2 Query Documents. .... 20
  - 4.3 Update Documents. .... 21
  - 4.4 Delete Documents..... 21

## 1. ¿Qué es MongoDB?

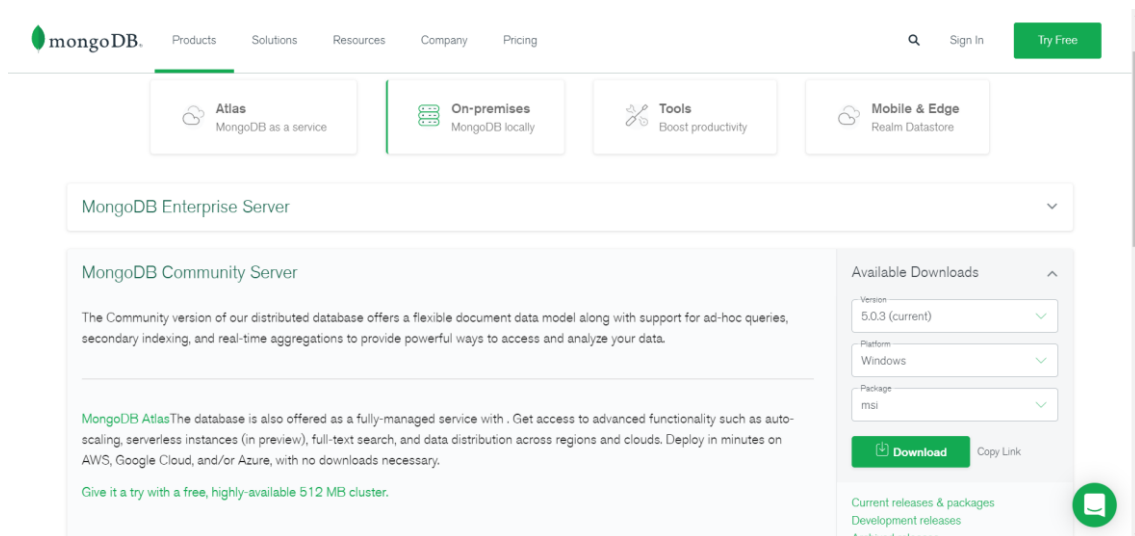
Es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

## 2. Instalacion Mongo:

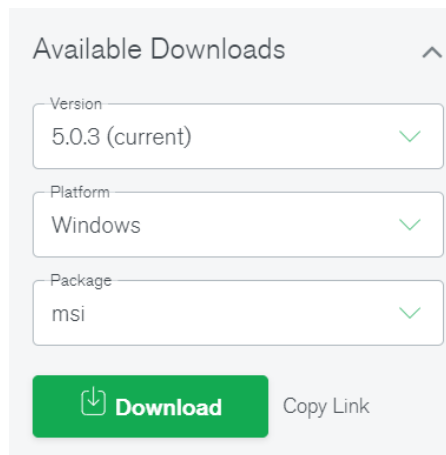
Vamos a la página de MongoDB, le daremos a Software e iremos a la primera opción, "Community Server".



Nos redirigirá a la siguiente página. Podemos ver que estamos en el apartado On-premises.



Podemos ver que podemos seleccionar la versión que deseamos descargar, para la plataforma que lo queremos usar y el tipo de archivo, en mi caso lo dejare todo predefinido y le daremos a download.




Available Downloads

Version  
5.0.3 (current) ✓

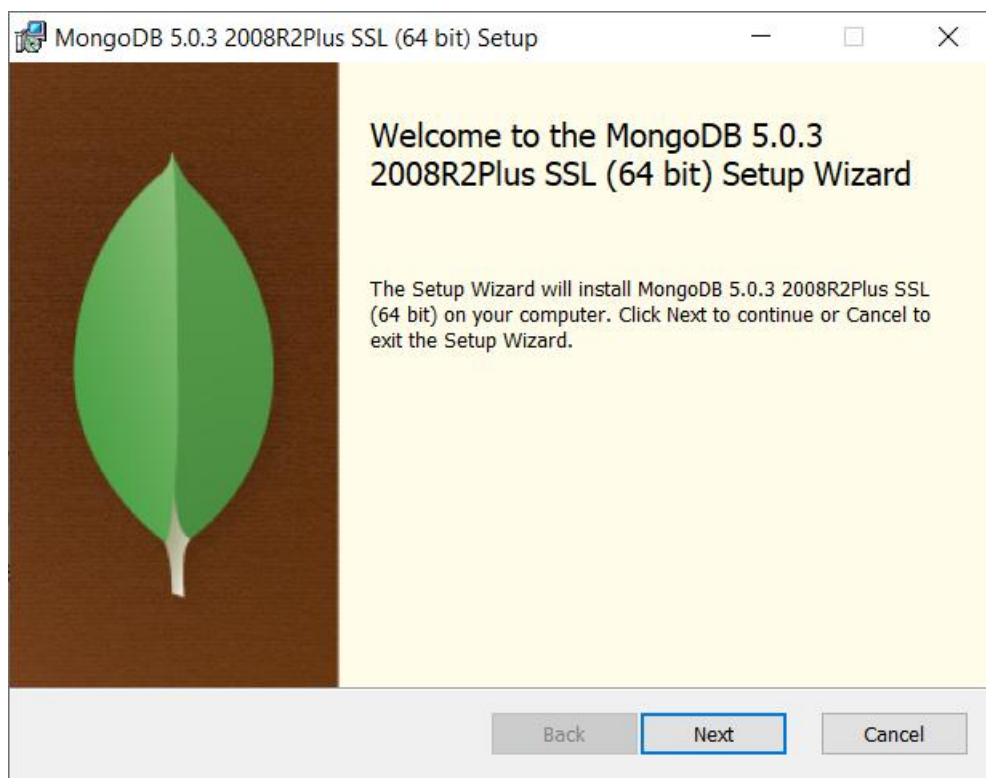
Platform  
Windows ✓

Package  
msi ✓

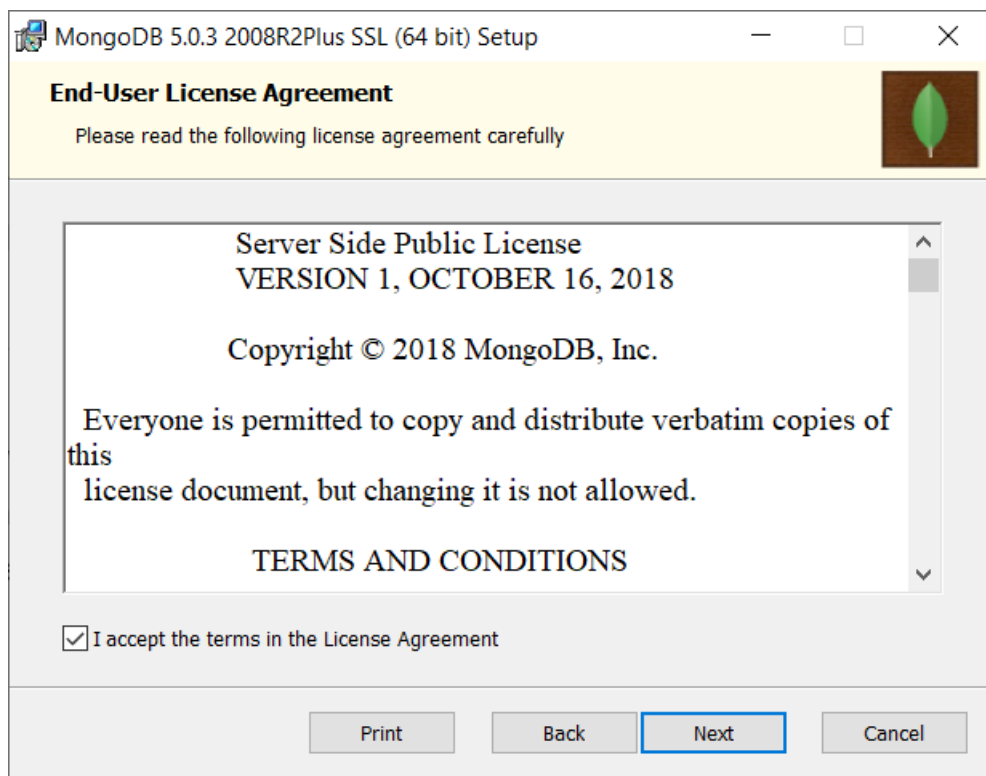
 **Download** [Copy Link](#)

Una vez descargado el instalador, lo ejecutaremos.

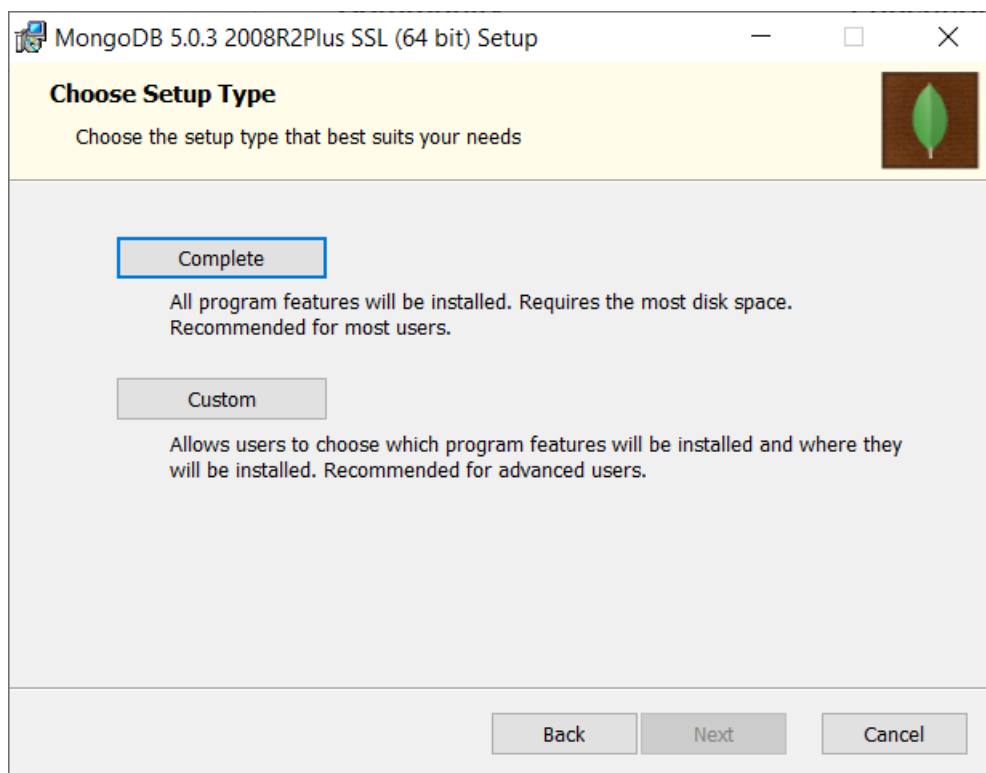
Le daremos a Next.



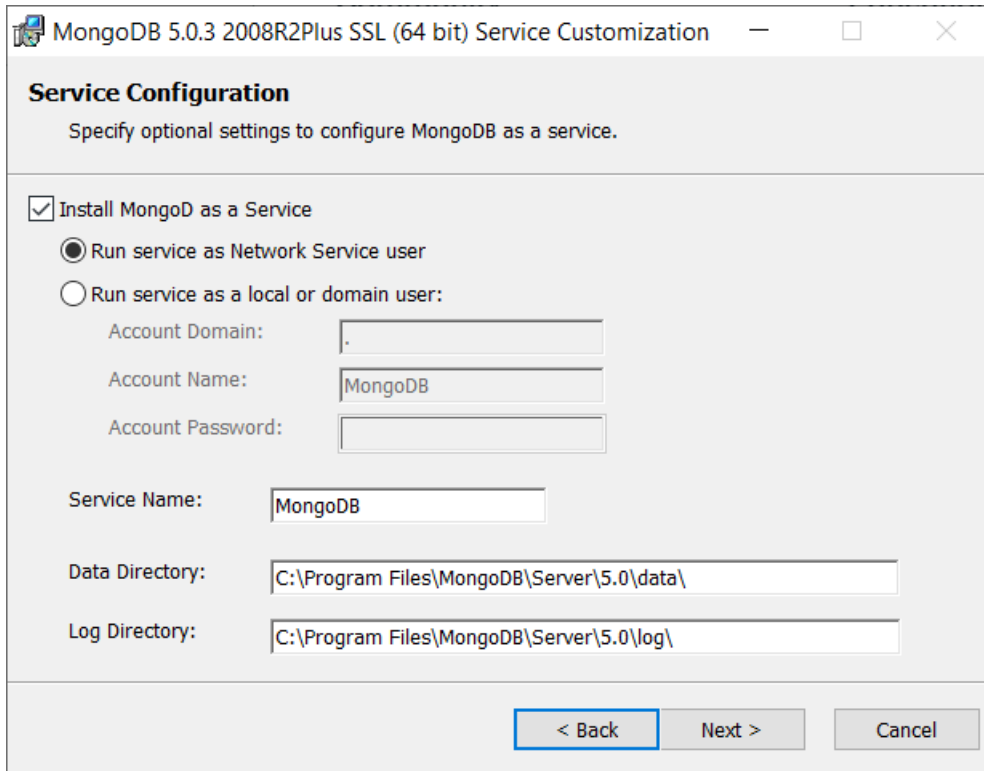
Aceptamos los términos y Next.



Le daremos a Complete y Next.

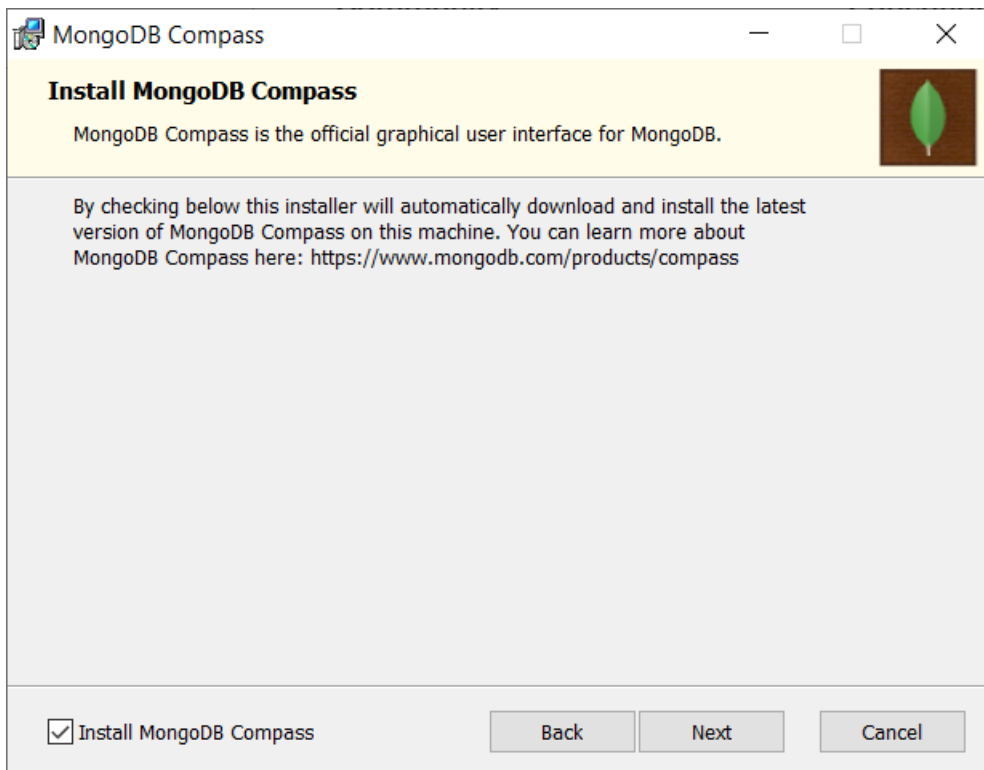


Dejaremos toda la configuración por defecto, hay que tener en cuenta que mongo se va a instalar como un servicio.



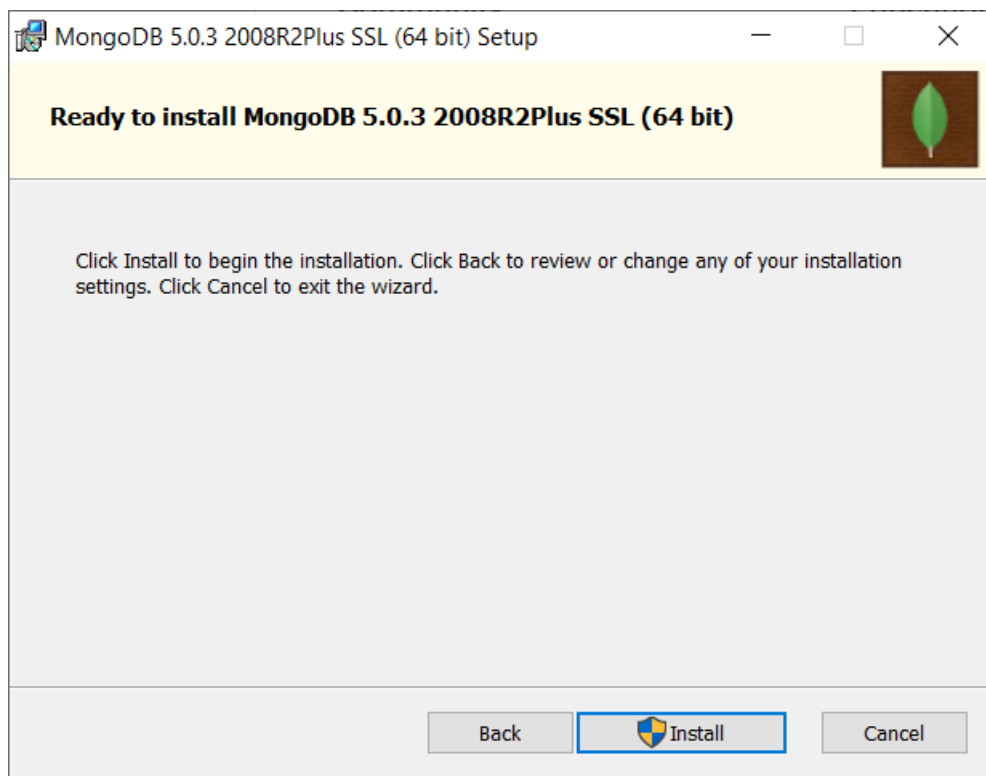
The screenshot shows the 'Service Configuration' window for MongoDB 5.0.3. The title bar reads 'MongoDB 5.0.3 2008R2Plus SSL (64 bit) Service Customization'. The main heading is 'Service Configuration' with the subtitle 'Specify optional settings to configure MongoDB as a service.' Below this, there are several configuration options: a checked checkbox for 'Install MongoDB as a Service', two radio buttons for 'Run service as Network Service user' (selected) and 'Run service as a local or domain user:', and three text input fields for 'Account Domain:', 'Account Name:' (containing 'MongoDB'), and 'Account Password:'. Further down are text input fields for 'Service Name:' (containing 'MongoDB'), 'Data Directory:' (containing 'C:\Program Files\MongoDB\Server\5.0\data\'), and 'Log Directory:' (containing 'C:\Program Files\MongoDB\Server\5.0\log\'). At the bottom right are three buttons: '< Back' (highlighted with a blue border), 'Next >', and 'Cancel'.

Hay que tener cuidado en este apartado, ya que mongo no funciona sin otro programa adicional, podemos ver abajo un apartado donde instalara MongoDB Compass, con esta aplicación interactuaremos con mongo. En caso de tener Mongo Compass, quitaremos el tick. Luego daremos a Next.

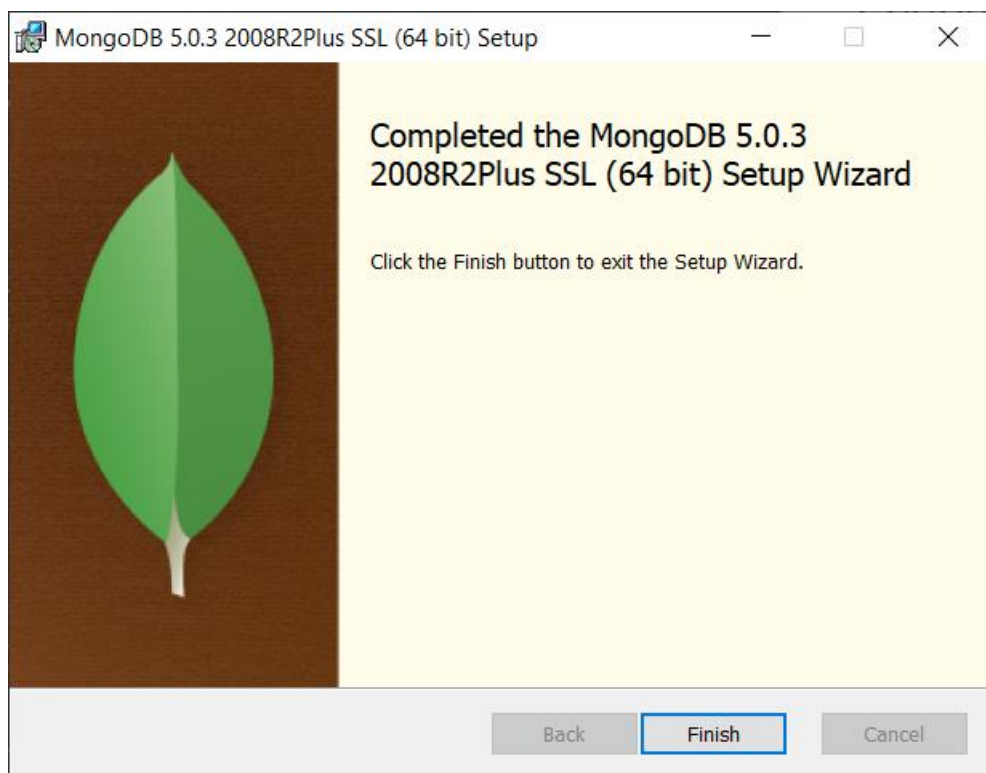


The screenshot shows the 'Install MongoDB Compass' window. The title bar reads 'MongoDB Compass'. The main heading is 'Install MongoDB Compass' with the subtitle 'MongoDB Compass is the official graphical user interface for MongoDB.' Below this is a paragraph of text: 'By checking below this installer 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/compass>'. At the bottom left is a checked checkbox for 'Install MongoDB Compass'. At the bottom right are three buttons: 'Back', 'Next', and 'Cancel'.

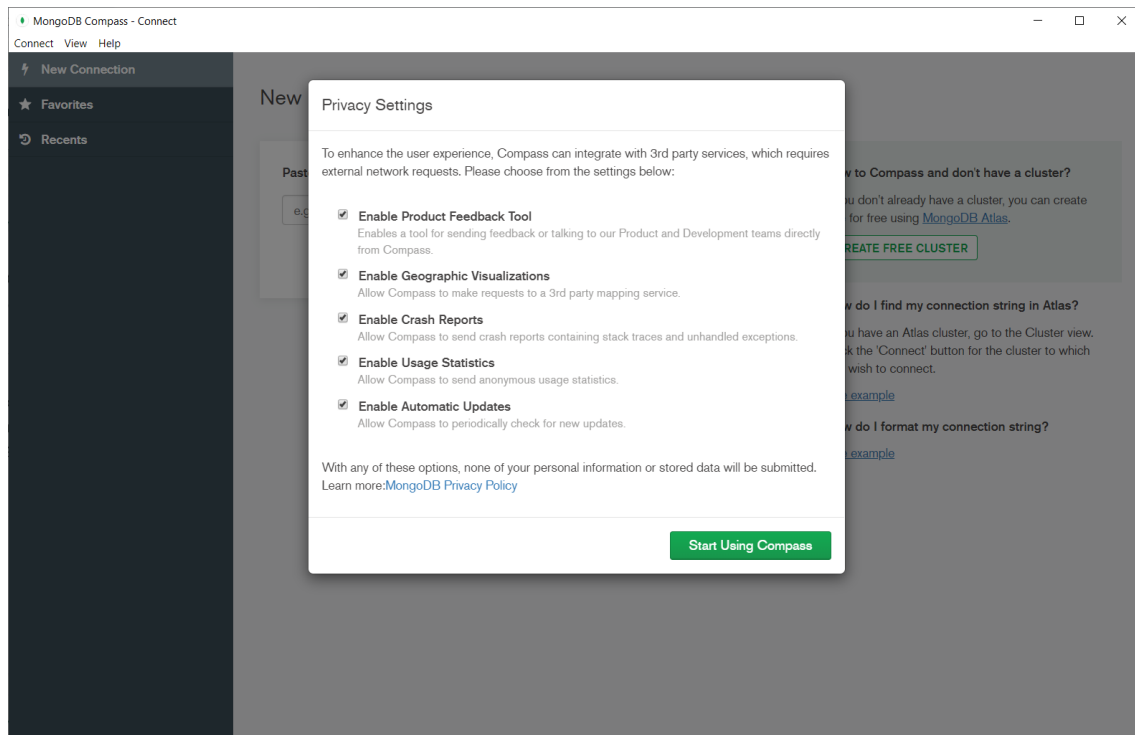
Lo siguiente que queda es comenzar a instalar, le daremos a Install.



Ya termino la instalación de MongoDB.

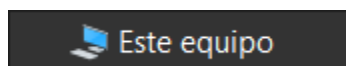


Una vez instalado se nos iniciara mongo compass, en este apartado dejaremos todos los tick seleccionados y le daremos a Start Using Compass.

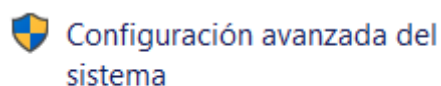


Es posible que PowerShell no detecte a mongo, haremos lo siguiente para poder trabajar con el.

Nos iremos a Explorador de Archivos y le daremos click derecho y propiedades en:

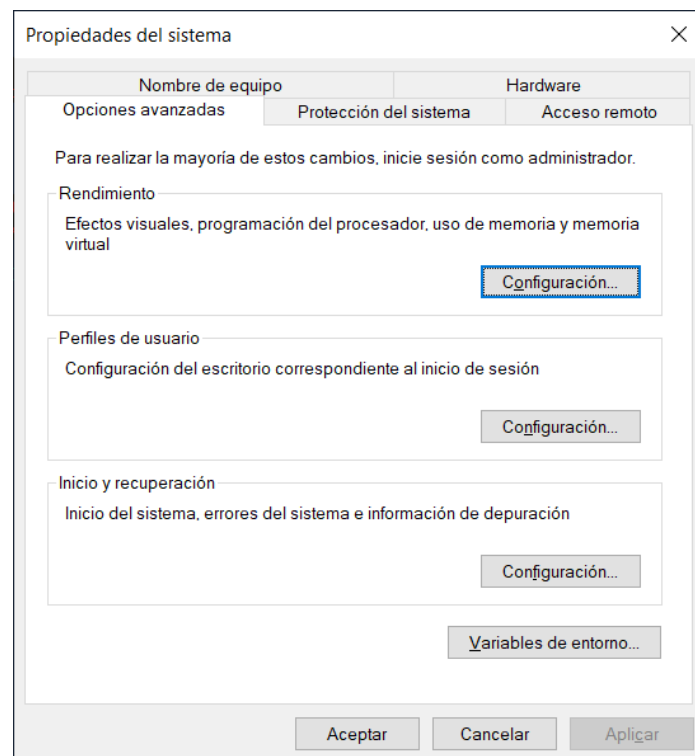


Ahora nos iremos al menú de la izquierda y le daremos a:

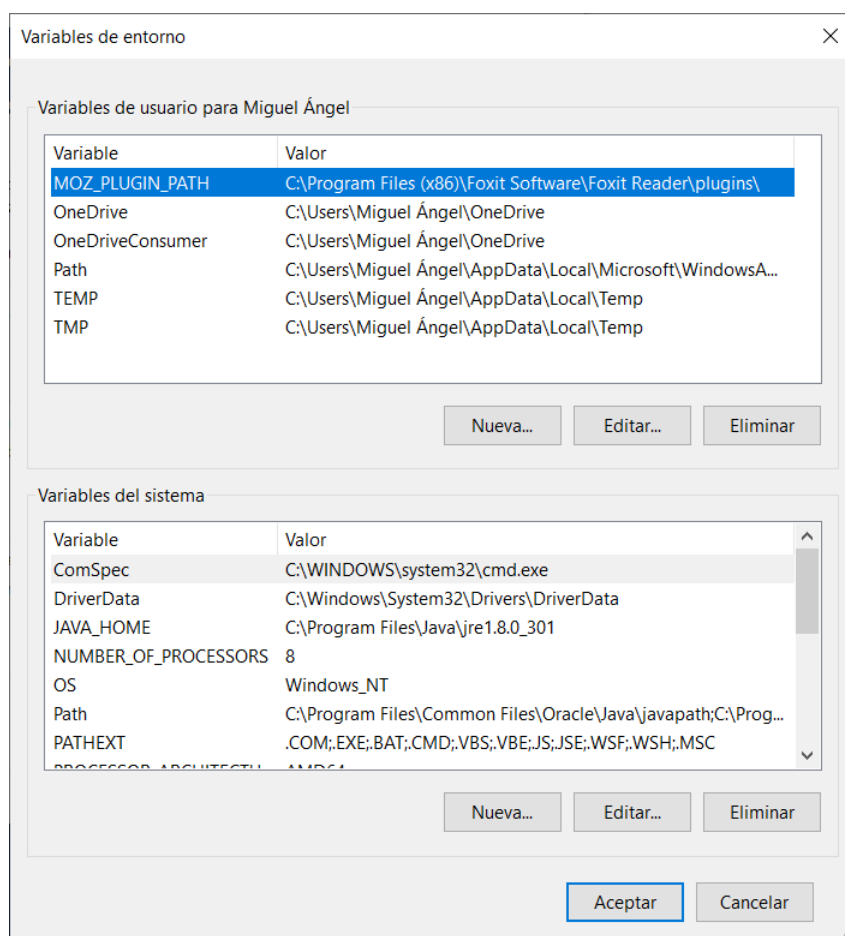




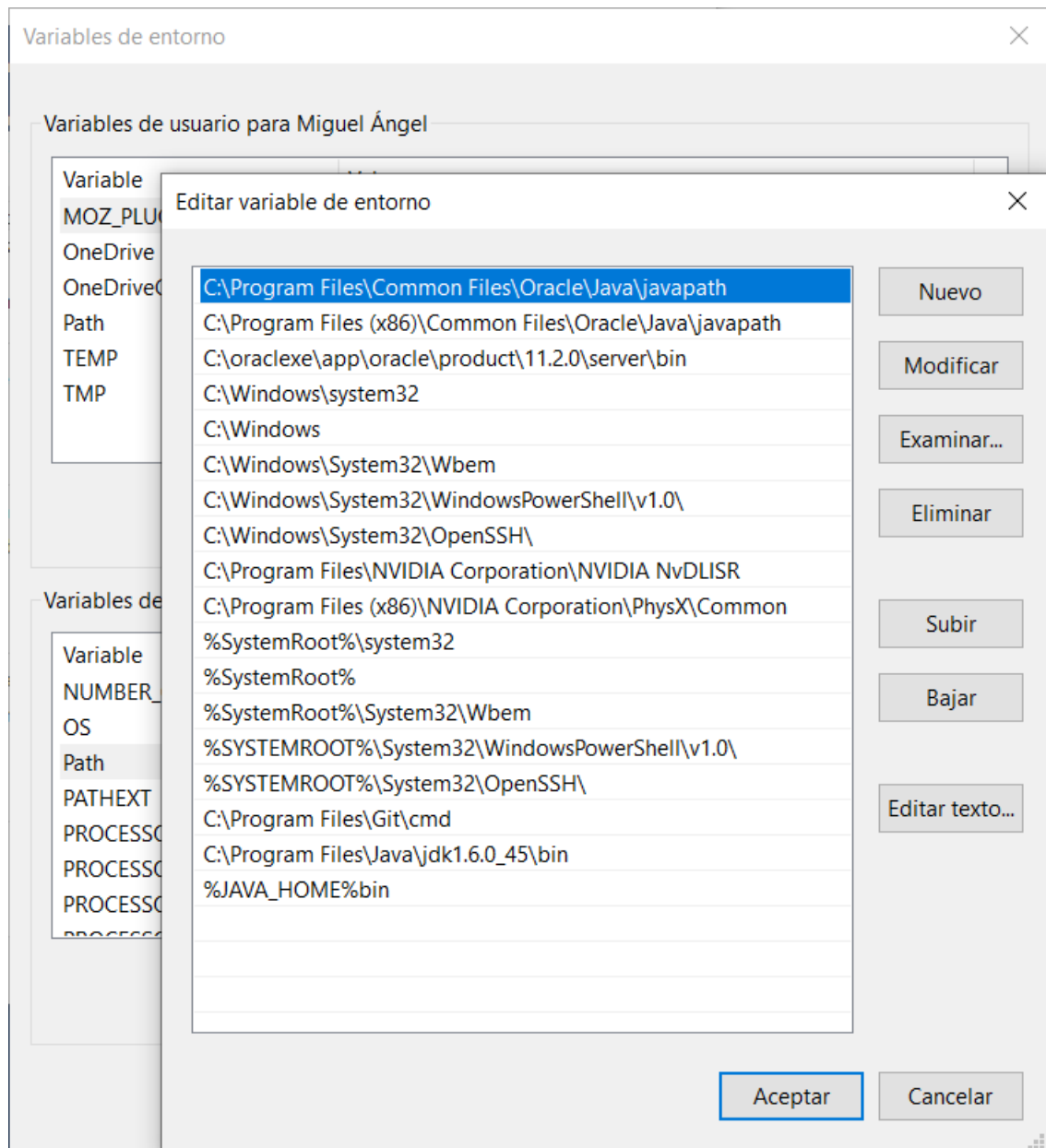
Una vez aquí, iremos a variables de entorno.



Ahora nos iremos al recuadro de abajo y buscaremos Path y le daremos a editar.



Le daremos a Nuevo



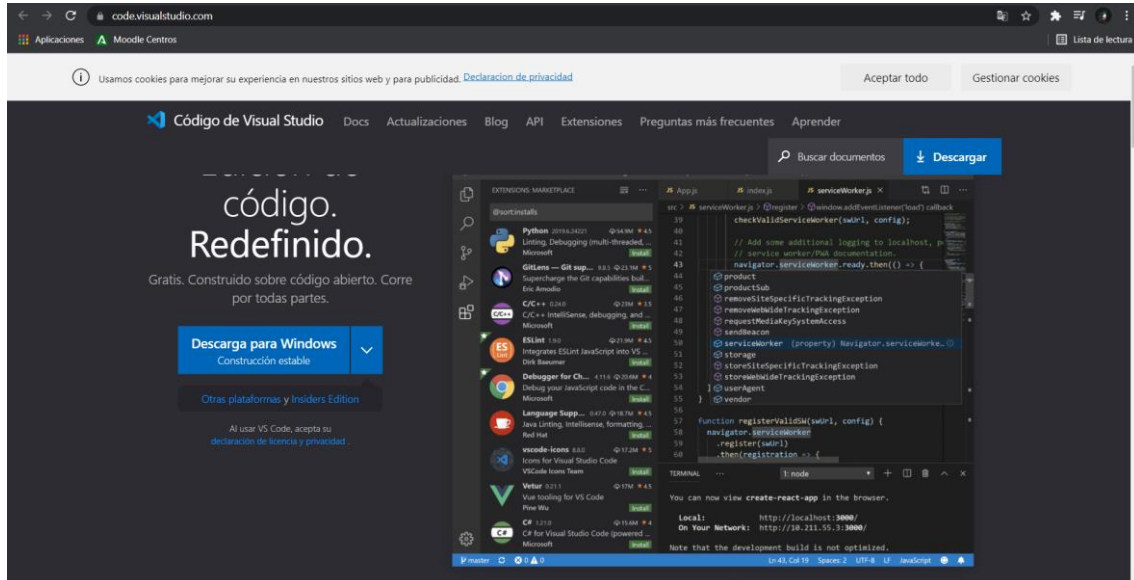
Iremos a donde se instaló MongoDB e iremos a la carpeta bin y copiaremos la ruta:

C:\Program Files\MongoDB\Server\5.0\bin

La ruta la tendremos que añadir en el Path y aceptaremos.

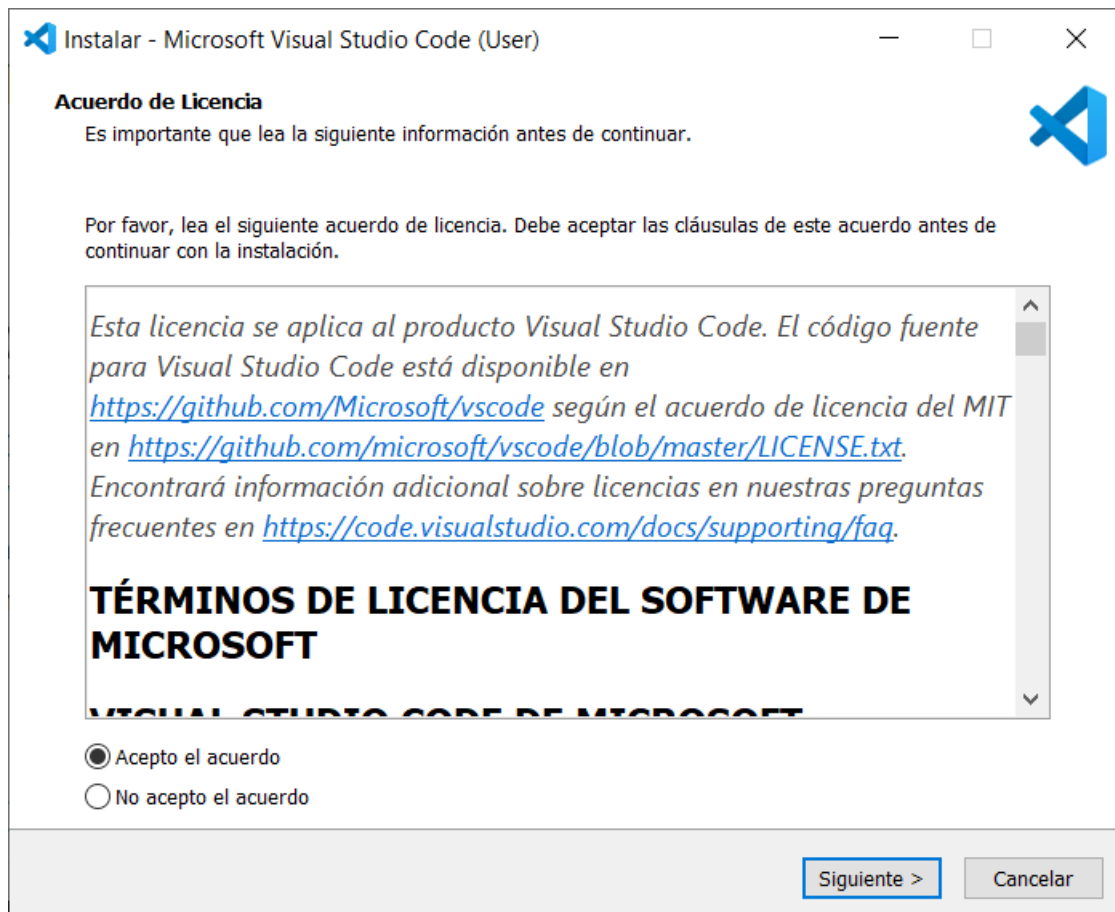
### 3. Instalación Visual Studio Code.

Nos iremos a la página de Visual Studio Code y le daremos a descargar.

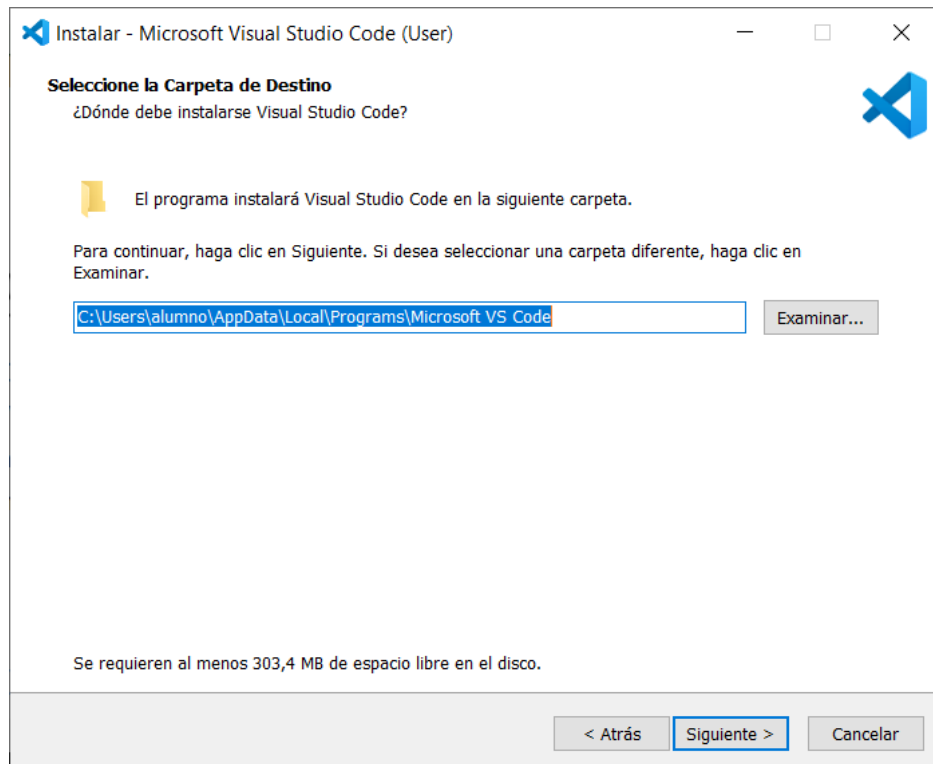


Una vez descargado el instalador lo ejecutaremos.

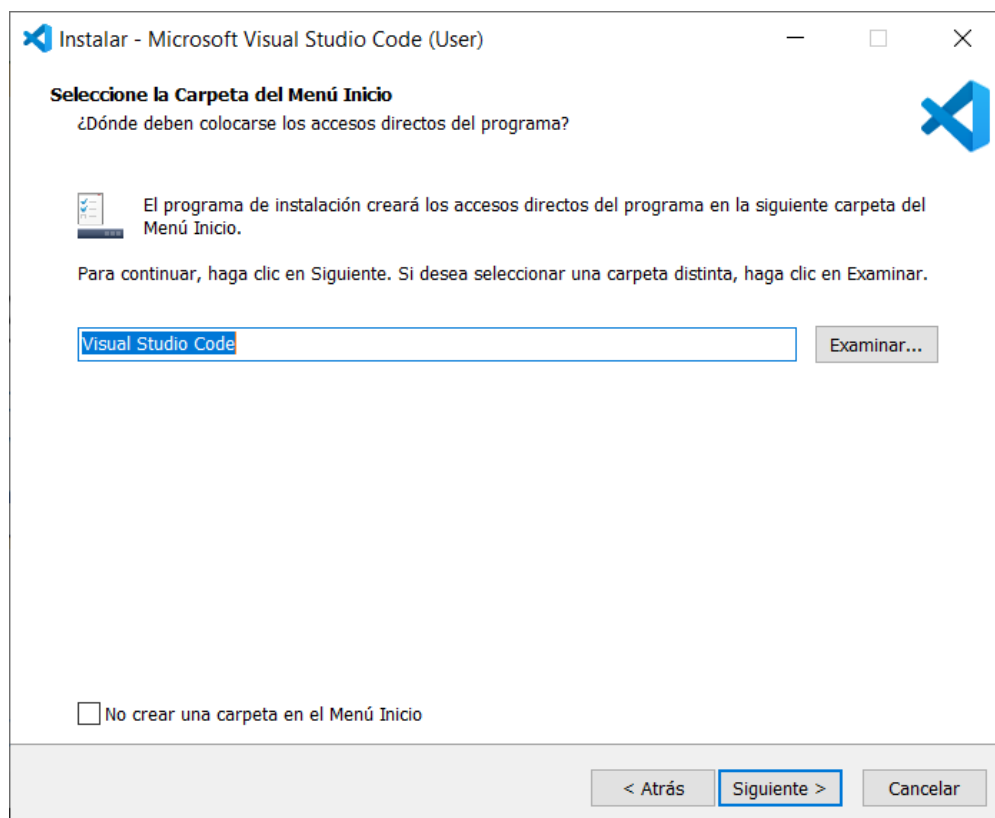
Nos saldrá una primera página con los términos, los aceptamos y siguiente.



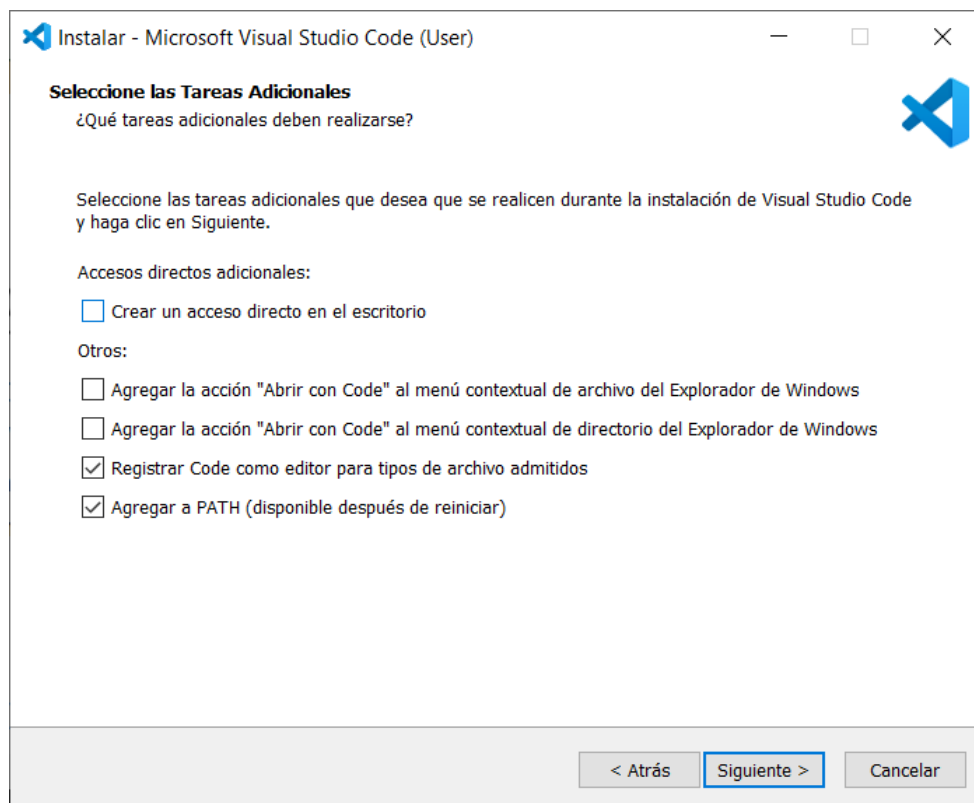
Seleccionamos donde instalaremos el programa, en nuestro caso, lo dejamos igual y le damos a siguiente.



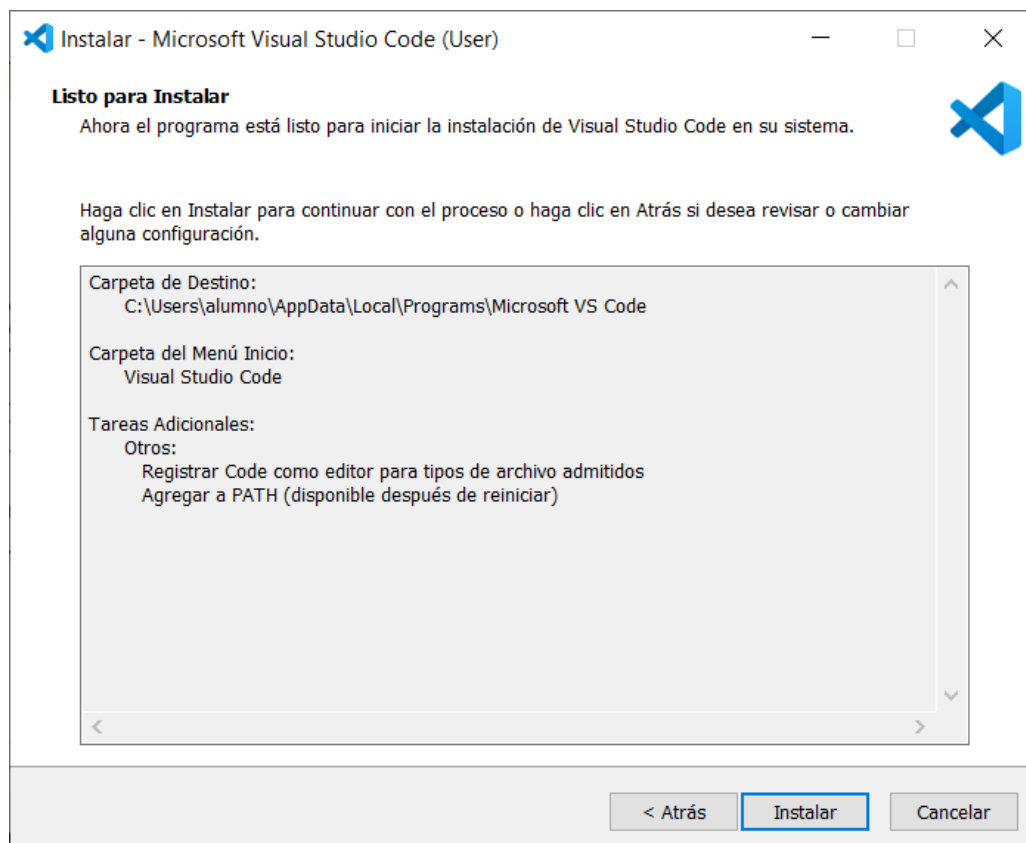
Le daremos a siguiente.



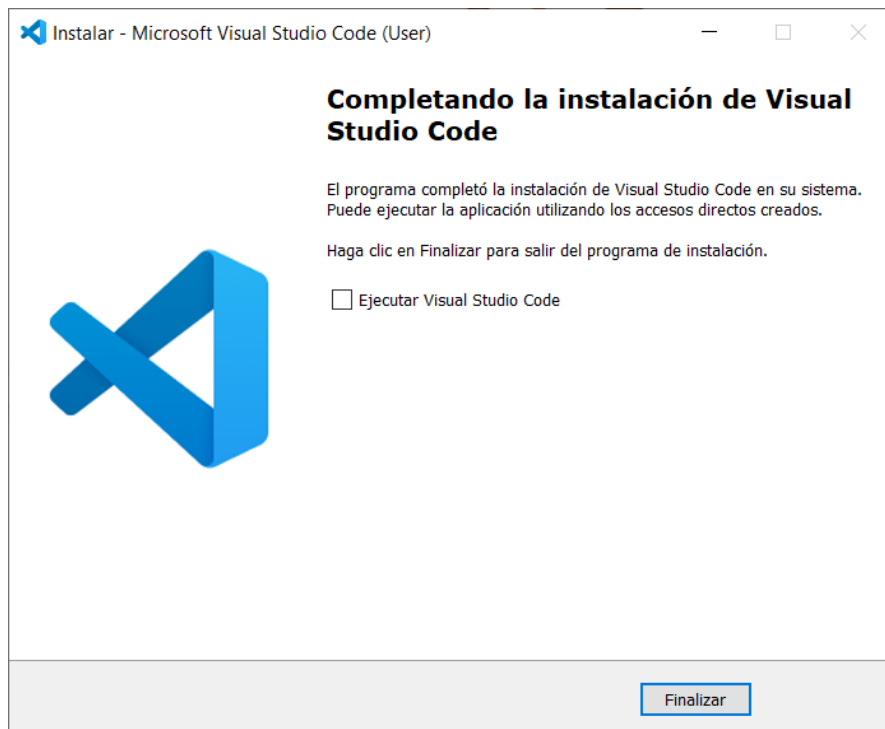
Solo marcaremos los 2 últimos ticks y le daremos a siguiente.



Le daremos a instalar.



Podemos ver que ya termino de instalarse.



### 3.1 Creación de un proyecto desde Visual Studio Code.

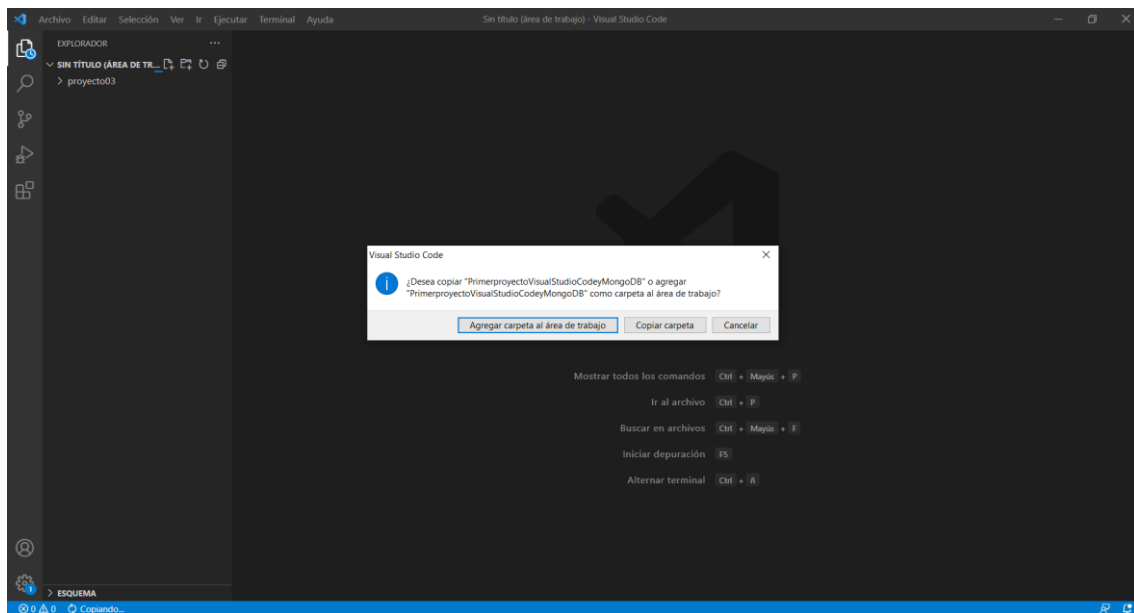
Crearemos un proyecto nuevo, en mi caso será proyecto02 y lo guardare en la siguiente ruta:

C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerproyectoVisualStudioCodeyMongoDB

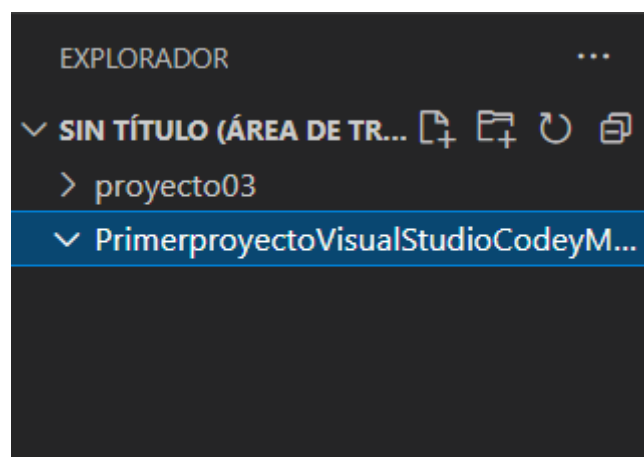
A continuación, iniciaremos Visual Studio, en caso de no tener la carpeta PrimerproyectoVisualStudioCodeyMongoDB en el explorador, podemos añadirla de 2 maneras:

1. Arrastrando desde el explorador de archivos a la aplicación.
2. Buscar la carpeta desde el explorador de la aplicación.

Nos saldrá el siguiente mensaje donde le daremos a Yes.



Ya tendremos el PrimerproyectoVisualStudioCodeyMongoDB en el menú de Visual Studio.

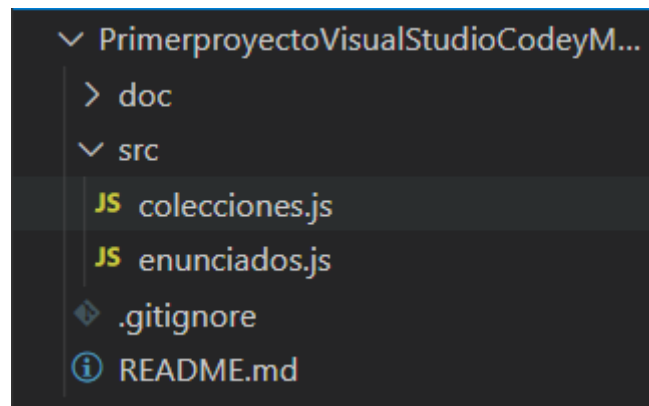


Podemos ver que está vacía, comencemos a crear la estructura del proyecto.

Crearemos la carpeta de src con unos archivos .js el cual usaremos más adelante, crearemos una carpeta doc, este tendrá el manual de la tarea, crearemos un archivo .gitignore (en este archivo pondremos el nombre de archivos o carpetas que no queremos que se suba a GitHub) y un README.md.

Si en algún momento creamos un archivo o carpeta en un sitio incorrecto, es tan fácil como arrastrarlo a la ubicación deseada.

Nos quedara la siguiente estructura:



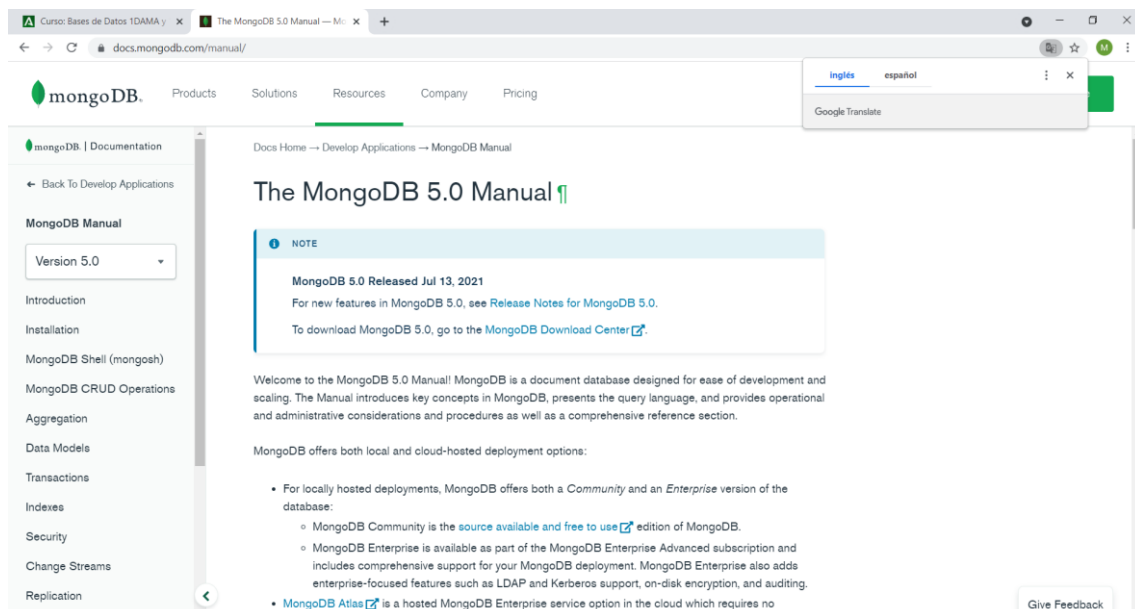
## 3.2 Antes de comenzar

Aquí buscaremos alguna de la información necesaria para hacer la practica:

En la página principal de mongoDB tenemos un apartado donde tenemos unas “Instrucciones” de cómo se usa mongo y los comandos que usa, lo podemos buscar en el panel del menú superior de la página Web buscando el apartado Docs. Luego le daremos a Server.

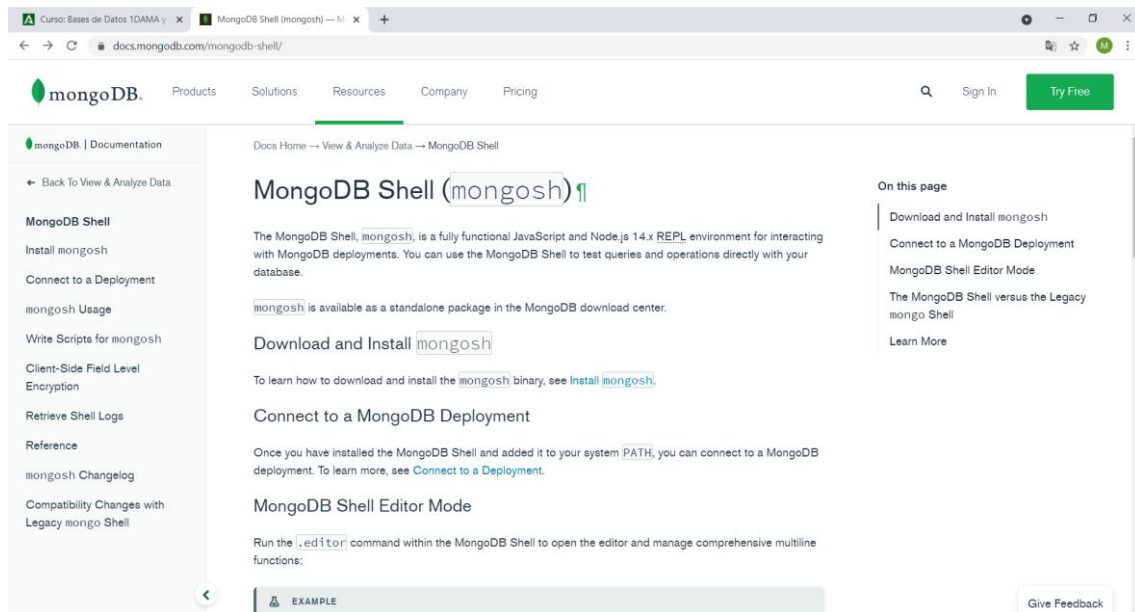


Podemos ver que tenemos muchos artículos de mongo.

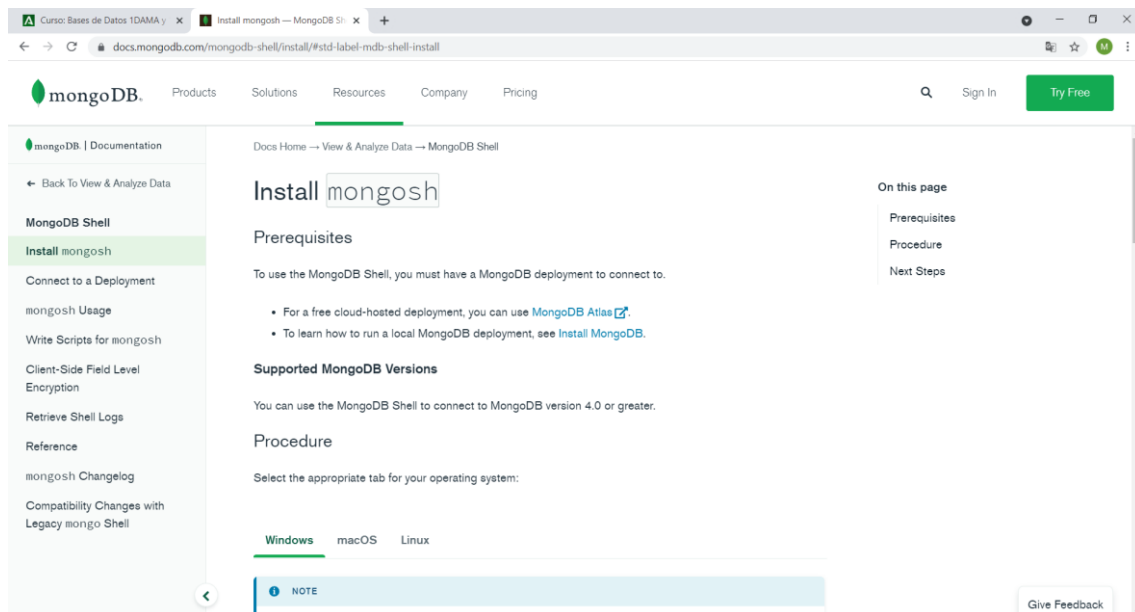


Si nos vamos al apartado de MongoDB Shell podemos ver que nos saldrán muchos comandos de los cuales iremos utilizando algunos a medida que sigamos con la explicación.

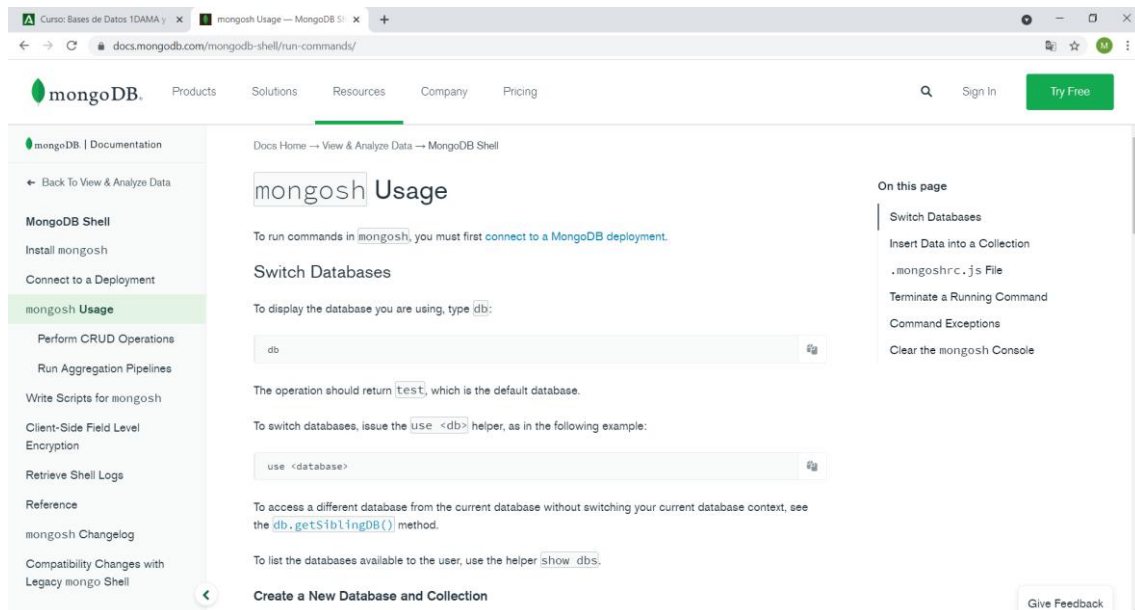




Mongo nos instala un Shell para trabajar con él desde PowerShell pero hace poco mongo ha sacado una nueva herramienta llamada mongosh, se podría decir es hace lo mismo pero esta actualizado con nuevas funciones. En caso de querer instalar este nuevo Shell nos iremos a Download and Install que podemos ver en la imagen anterior. Luego nos redirigirá a un nuevo sitio y podemos ver en el segundo punto Install MongoDB. En nuestro caso seguiremos usando el Shell que nos trae por defecto.



Volvemos a MongoDB Shell y nos vamos a mongodb Usage, aquí veremos algunos comandos interesantes.



Nos vamos a PowerShell y pondremos a prueba algunos comandos.

Para entrar con mongo es tan fácil como posicionarte en la carpeta donde quieras trabajar y escribir mongo.

Para comenzar, haremos un db y nos devolverá la Base de Datos que estamos usando.

Si usamos use le diremos la Base de datos que queremos utilizar.

```
> db
test
> use bdp01
switched to db bdp01
>
```

Aunque hayamos cambiado de Base de Datos, aún no está creada, tendremos que meterle datos para crear así la estructura.

Ahora introduciremos algunos datos, usaremos el siguiente comando:

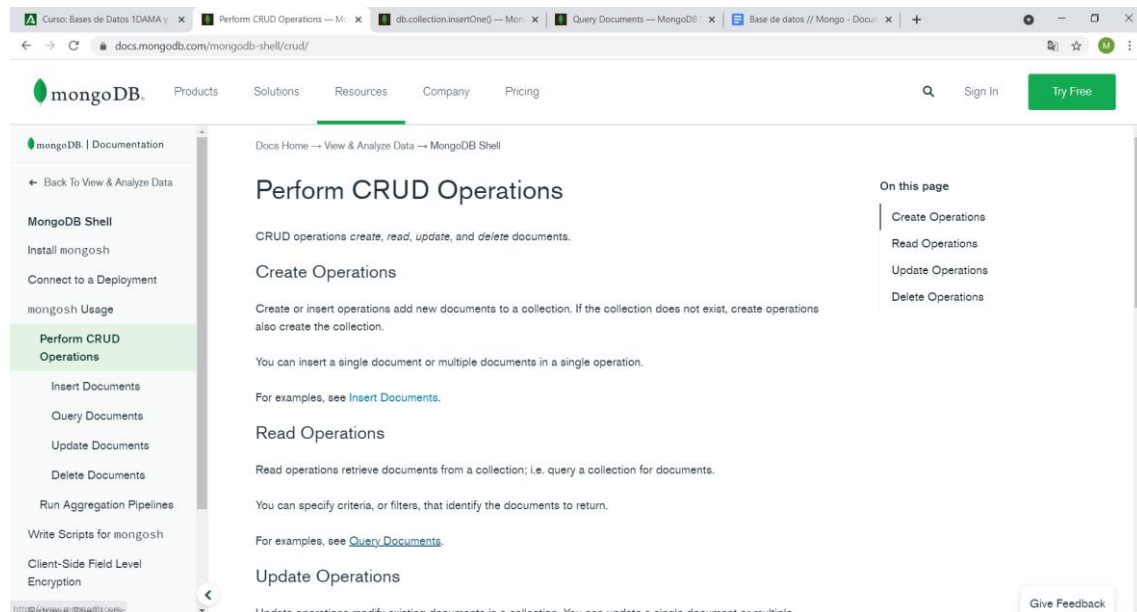
```
> db
bdp01
> db.col01.insertOne({nombre: "Pepe", edad: 19 })
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61555ec7dcf4ebad08f22faa")
}
>
```

Una Base de Datos está compuesta por: db → Collections → Documents.

Por eso en el comando anterior podemos ver db(Base de Datos que estamos usando actualmente), col01(Colección que estamos creando que contendrá los datos que vamos a meterle) y el comando donde meterá los datos de uno en uno.

Se le puede meter más de un dato igual ya que se diferencian entre ellos por ID.

Otro apartado de la Web de interés es para crear, modificar y eliminar cosas, podemos verlo en Perform CRUD Operations.



Si usamos el siguiente comando podremos ver que contiene la colección que queramos.

db(Base de Datos), col01 (Colección) y find para mostrar.

```
> db.col01.find()
{ "_id" : ObjectId("61555ec7dcf4ebad08f22faa"), "nombre" : "Pepe", "edad" : 19 }
>
```

Podemos saber cuántas Bases de Datos tenemos introduciendo:

```
> show dbs
admin    0.000GB
bdp01    0.000GB
config   0.000GB
local    0.000GB
>
```

De la misma manera podemos ver cuántas colecciones tenemos (de la Base de Datos que estamos usando)

```
> show collections
col01
>
```

## 4. Insert, Query, Update y Delete.

### 4.1 Insert Documents.

Vamos a colocarnos en la carpeta de la practica desde el PowerShell y luego entraremos en mongo. Crearemos una base de datos para la practica.

```
> db
test
> use PracticaVisualyMongo
switched to db PracticaVisualyMongo
> db
PracticaVisualyMongo
> |
```

Una vez que estamos en la db con la que vamos a trabajar, esta no está creada, se creara cuando le creamos una colección con la que trabajar.

Así que vamos a crear la colección:

```
> db.createCollection("practica")
{ "ok" : 1 }
```

De este modo, se ha creado el db y la colección:

```
> show dbs
PracticaVisualyMongo 0.000GB
admin                 0.000GB
bdp01                 0.000GB
config                0.000GB
local                 0.000GB
test                  0.000GB
> show collections
practica
```

Iremos al archivo donde introduciremos un comando para introducir datos.

Inserta solo 1 documento.

```
JS colecciones.js JS enunciados.js X
PrimerproyectoVisualStudioCodeyMongoDB > src > JS enunciados.js > ...
1 db.practica.insertOne( { Nombre: "Miguel Angel" }, )
```

Inserta muchos documentos.

```
db.practica.insertMany([
  // objeto          cantidad          Podríamos decir que es un localizador
  { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
  { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
  { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
  { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
  { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" },
])
```

Elimina todo el contenido de la colección. Si delante tiene // quiere decir que esta como comentario.

```
// db.practica.deleteMany({})
```

Vamos a introducir datos desde el archivo enunciados.js:

```
> load("enunciados.js")
true
```

## 4.2 Query Documents.

Vamos a verificar que se han introducido correctamente los datos. Si usamos una estructura para introducir los datos, a la hora de introducirlo se verán de recorrido y no con la estructura que teníamos, en todo caso usaríamos el comando find y de seguido .pretty().

```
> db.practica.find()
{ "_id" : ObjectId("616b172dcb3cb4122a262731"), "Nombre" : "Miguel Angel" }
{ "_id" : ObjectId("616b172dcb3cb4122a262732"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("616b172dcb3cb4122a262733"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("616b172dcb3cb4122a262734"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("616b172dcb3cb4122a262735"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("616b172dcb3cb4122a262736"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
>
```

Podemos ver que se ha creado correctamente, también podemos buscar datos por ejemplo por el apartado de status y letra D.

```
> db.practica.find({"status": "D"})
{ "_id" : ObjectId("616b172dcb3cb4122a262734"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("616b172dcb3cb4122a262735"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
```

Podemos ver que una vez introducimos datos, podemos buscarlos fácilmente filtrándolos como por ejemplo, en la imagen anterior.

También podemos decirle que solo nos muestre un dato, por ejemplo size.

```
> db.practica.find({"status": "D"}, {size: 1})
{ "_id" : ObjectId("616b172dcb3cb4122a262734"), "size" : { "h" : 8.5, "w" : 11, "uom" : "in" } }
{ "_id" : ObjectId("616b172dcb3cb4122a262735"), "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" } }
```

La \_id sale por defecto pero si queremos quitarlo solo tendremos que poner {size: 1, \_id:}

También podemos usar .sort si meterle alguna de las casillas para ordenarlo alfabéticamente.

Si usamos db.practica.find().limit() y dentro de limit podemos decirle cuantos datos queremos que nos muestre.

Si usamos db.practica.count() nos muestra de forma numérica cuantos datos hay introducidos.

## 4.3 Update Documents.

Hay 2 formas de actualizar los campos de nuestro documento, eliminando todo el contenido y sustituyéndolo por el nuevo o agregando el contenido nuevo al que ya tenemos.

Comenzaremos con la primera en la que se eliminara el contenido.

Haremos un find para localizar a notebook, podemos ver todo su contenido. Ahora le haremos un update con un precio y podemos ver que desaparece todo su contenido y aparecerá precio: 500. La sintaxis con la que hemos cambiado este dato es muy simple, entre los primeros {} pondremos el dato que queremos modificar y entre los segundos {} pondremos lo que queremos añadir para sustituir el contenido.

```
> db.practica.find({"item": "notebook"})
{ "_id" : ObjectId("616b172dcb3cb4122a262733"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
> db.practica.update({"item": "notebook"}, {"precio": 500.00})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.practica.find()
{ "_id" : ObjectId("616b172dcb3cb4122a262731"), "Nombre" : "Miguel Angel" }
{ "_id" : ObjectId("616b172dcb3cb4122a262732"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("616b172dcb3cb4122a262733"), "precio" : 500 }
{ "_id" : ObjectId("616b172dcb3cb4122a262734"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("616b172dcb3cb4122a262735"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("616b172dcb3cb4122a262736"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
```

Ahora vamos a introducir datos sin eliminar el contenido que hay dentro, lo haremos de la siguiente manera:

```
> db.practica.find({"precio" : 500})
{ "_id" : ObjectId("616b172dcb3cb4122a262733"), "precio" : 500 }
> db.practica.update({"precio" : 500}, {$set: {"nuevo dato": "television"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.practica.find({"precio" : 500})
{ "_id" : ObjectId("616b172dcb3cb4122a262733"), "precio" : 500, "nuevo dato" : "television" }
```

La sintaxis es muy sencilla, en los primeros {} ponemos el dato del que queremos añadirle los datos, en el segundo {} podemos ver que contiene \$set: , este lo que hace es añadir los datos que tendrá dentro de otro {} que tiene de seguido.

Podemos comprobar en la imagen anterior que se han introducido los datos correctamente.

## 4.4 Delete Documents.

Ahora vamos a ver como eliminar documentos, en el caso de que queramos eliminar todo el contenido de db.practica, es tan fácil como poner db.practica.remove({}) y se eliminara todo el contenido.

Vamos a ver también como eliminar db y collections, es bastante fácil, para eliminar una colección, en nuestro caso seria practica, es tan fácil como poner: db.practica.drop().

En el caso de que queramos borrar toda la db, es tan fácil como poner db.dropDatabase().