

Uso básico de Visual Studio Code y MongoDB

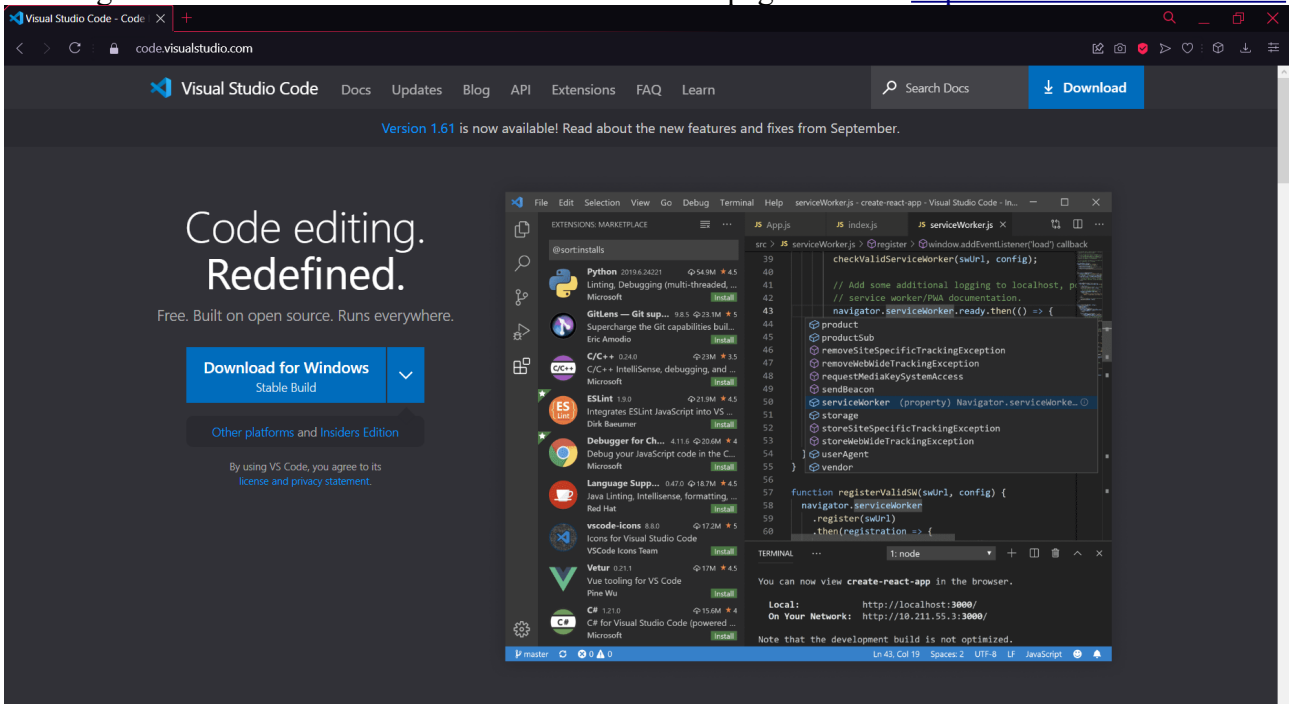
Iván Gasent Zarza

Indice

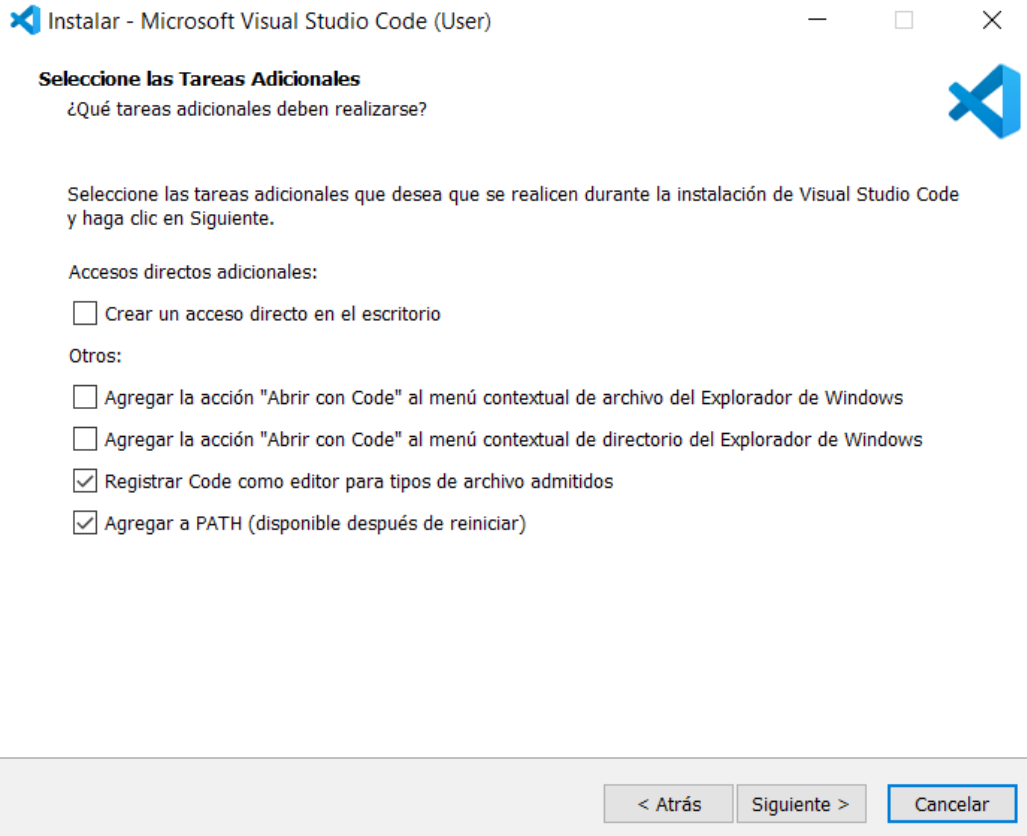
Indice	1
Instalación de Visual Studio Code	3
Instalación de MongoDB	3
Creación de bases de datos	5
Añadir documentos a una colección	5
Eliminar documentos de una colección	5
Leer documentos de una colección	6
Ejecución de scripts	6
Creación de scripts	7
Bibliografía	8

Instalación de Visual Studio Code

Descargue el instalador de Visual Studio Code desde su página oficial <https://code.visualstudio.com>

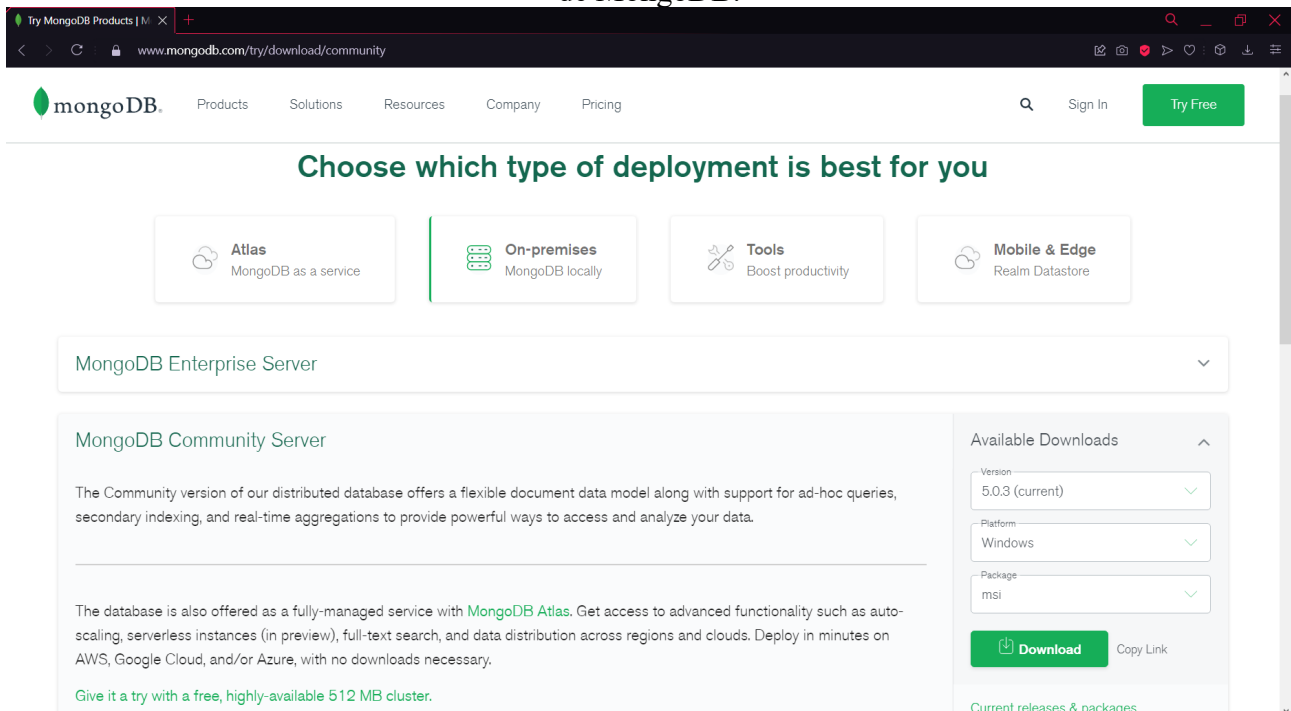


Ejecute el instalador cuando esté descargado y elija las opciones que sea conveniente.

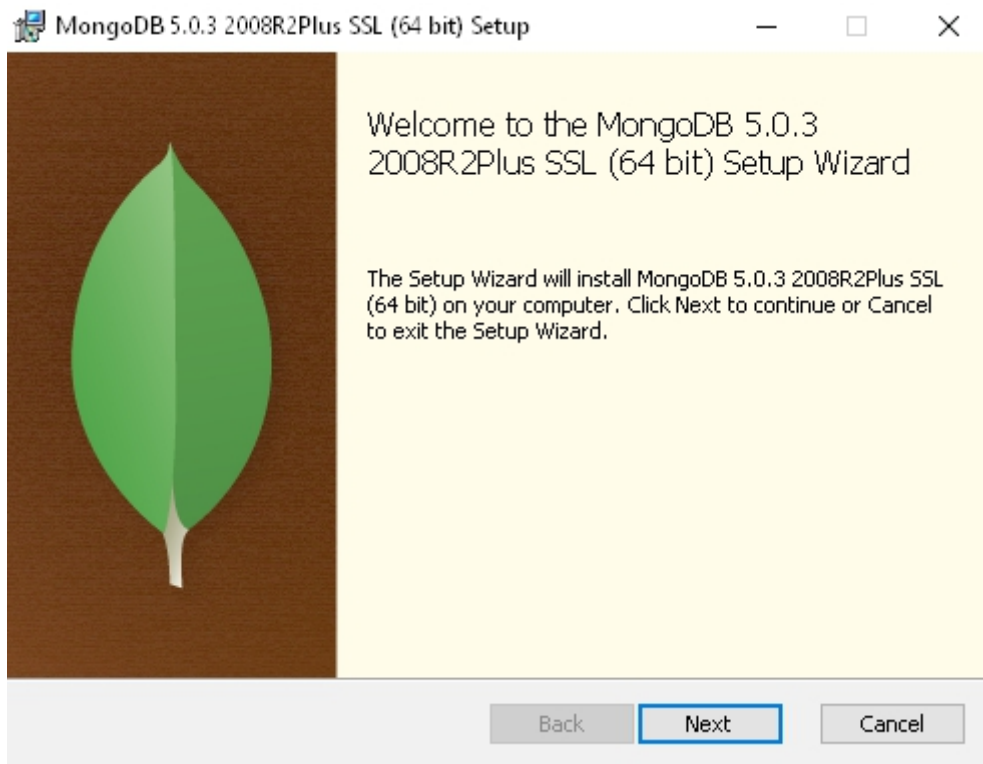


Instalación de MongoDB

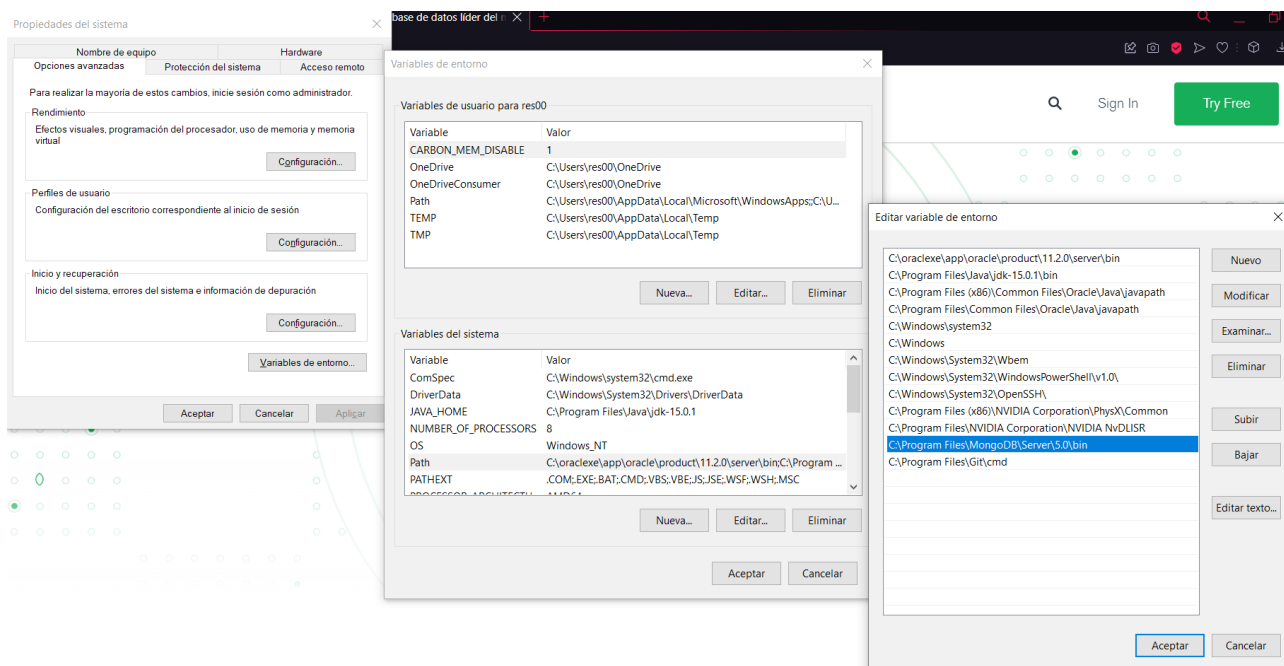
Entre en <https://www.mongodb.com/try/download/community> y descargue la Community Edition de MongoDB.



Ejecute el instalador y elija la instalación completa. Esto instalará MongoDB y Compass



Para poder usar los comandos de consola de mongo tendremos que crear una variable de entorno nosotros mismos. Para hacer esto acceda a variables del sistema, variables de entorno, seleccione Path en variables del sistema o creela si no existe, edítela y añada la siguiente dirección si seguiste los pasos por defecto C:\Program Files\MongoDB\Server\5.0\bin



Para comprobar que puedes usar mongo en PowerShell use:

mongo -version

Debería de aparecer lo siguiente.

```
PS C:\Users\res00> 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"
  }
}
```

Cuando quiera usar mongo en PowerShell solo tiene que escribir mongo.

```
PS C:\Users\res00> mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("8c950133-1e39-46d3-b131-79d07424e5f4") }
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:
2021-10-07T12:31:12.803+02:00: Access control is not enabled for the database. Read and write access 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()
-----
>
```

Creación de bases de datos

Por defecto, mongo empieza en la base de datos test, para cambiar a otra base de datos tienes que usar el comando use. Para ver la base de datos en la que te encuentras hay que usar el comando db y el comando show dbs muestra todas las bases de datos.

```
> db
test
> use bdprueba
switched to db bdprueba
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
```

La base de datos no se creará hasta que no tenga algún documento.

Añadir documentos a una colección

Para añadir un documento a una base de datos, tendremos que añadirla a una colección de esta. Para hacer esto se pueden usar los siguientes comandos que añaden uno o varios documentos a una colección:

```
db.NOMBRE_COLECCIÓN.insertOne()
db.NOMBRE_COLECCIÓN.insertMany()
```

```
> db.pruebacoleccion.insertOne({campo1:"valor1", campo2:23})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("6168961140a38cf7f0656061")
}
> db.pruebacoleccion.insertMany([{{campo1:"test", campo2:57},{campo1:"prueba",campo2:1},{campoUno:34,campoCuatro:"otro valor"}}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6168972240a38cf7f0656062"),
    ObjectId("6168972240a38cf7f0656063"),
    ObjectId("6168972240a38cf7f0656064")
  ]
}
```

Eliminar documentos de una colección

Para eliminar un documento, al igual que para añadir, tenemos un comando para eliminar uno y otro para eliminar varios. El parámetro a dar en estos comandos es un campo y valor, que eliminará a aquellos documentos que tengan ese valor en el campo. Si a deleteMany se le da como parámetro un campo valor vacío se eliminarán todos los documentos.

```
db.NOMBRE_COLECCIÓN.deleteOne()
db.NOMBRE_COLECCIÓN.deleteMany()
```

```
> db.pruebacoleccion.deleteOne({campo1:"valor1"})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.pruebacoleccion.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 3 }
```

Leer documentos de una colección

Para leer documentos se usará el comando `find`, que toma como parámetro de entrada campo y valor.

`db.NOMBRE_COLECCIÓN.find()`

```
> db.pruebacoleccion.insertMany([ { _id: 1, item: { name: "ab", code: "123" }, qty: 15, tags: [ "A", "B", "C" ] }, { _id: 2, item: { name: "cd", code: "123" }, qty: 20, tags: [ "B" ] }, { _id: 3, item: { name: "ij", code: "456" }, qty: 25, tags: [ "A", "B" ] }, { _id: 4, item: { name: "xy", code: "456" }, qty: 30, tags: [ "B", "A" ] }, { _id: 5, item: { name: "mn", code: "000" }, qty: 20, tags: [ [ "A", "B" ], "C" ] } ] )
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4, 5 ] }
> db.pruebacoleccion.find({qty:20})
{ "_id" : 2, "item" : { "name" : "cd", "code" : "123" }, "qty" : 20, "tags" : [ "B" ] }
{ "_id" : 5, "item" : { "name" : "mn", "code" : "000" }, "qty" : 20, "tags" : [ [ "A", "B" ], "C" ] }
```

Para preguntar por un valor que este compuesto por más campos valores:

```
> db.pruebacoleccion.find({"item.name":"cd"})
{ "_id" : 2, "item" : { "name" : "cd", "code" : "123" }, "qty" : 20, "tags" : [ "B" ] }
> db.pruebacoleccion.find({tags:"A"})
{ "_id" : 1, "item" : { "name" : "ab", "code" : "123" }, "qty" : 15, "tags" : [ "A", "B", "C" ] }
{ "_id" : 3, "item" : { "name" : "ij", "code" : "456" }, "qty" : 25, "tags" : [ "A", "B" ] }
{ "_id" : 4, "item" : { "name" : "xy", "code" : "456" }, "qty" : 30, "tags" : [ "B", "A" ] }
```

Si se pregunta por el valor de un array se responde a si hay un valor igual dentro del array.

Ejecución de scripts

Para ejecutar un script se debe usar el comando `load` con el nombre del archivo `.js` y haber abierto mongo en la carpeta donde se encuentre el archivo.

```
PS C:\Users\res00> cd C:\Users\res00\Documents\1DAM\proyecto02\src
PS C:\Users\res00\Documents\1DAM\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("62db9105-22ed-4ebe-a0c6-59e3639f577a") }
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:
  2021-10-07T12:31:12.803+02:00: Access control is not enabled for the database. Read and write access 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).

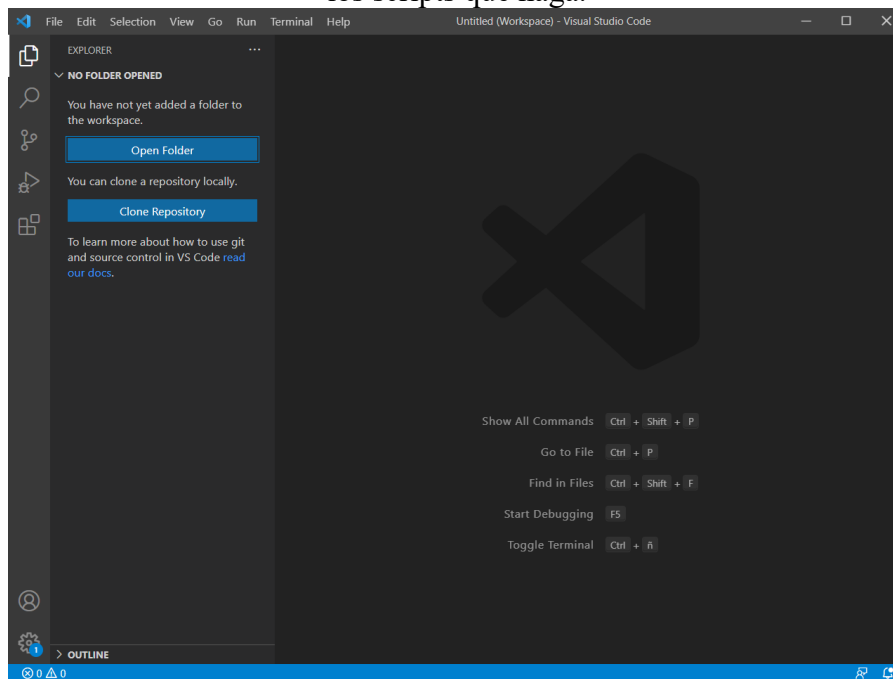
  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
  and anyone you share the URL with. MongoDB may use this information to make product
  improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> load("inserts01.js")
true
```

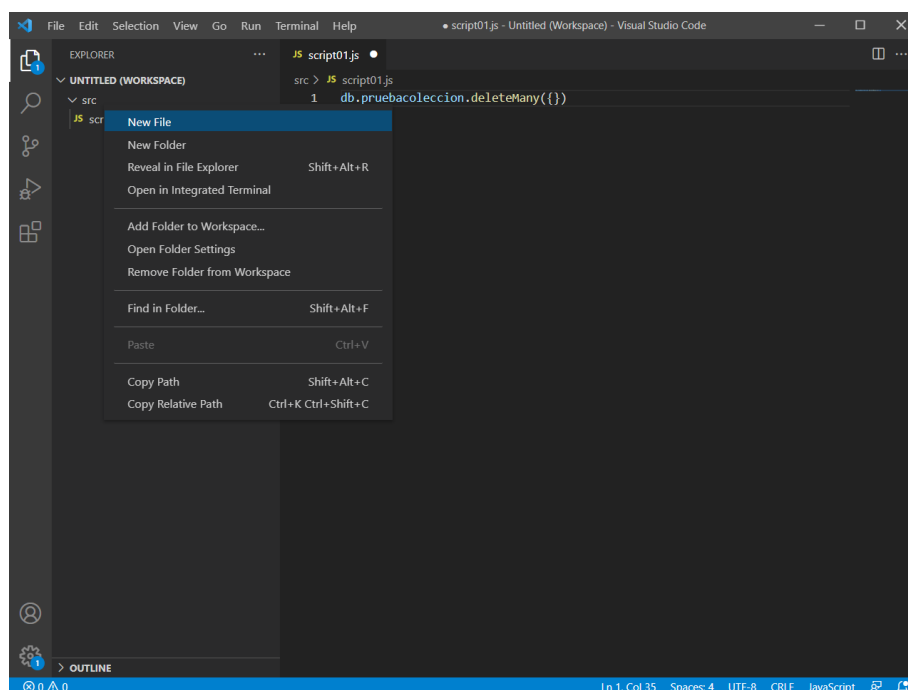
Creación de scripts

Para crear scripts solo hace falta crear un archivo .js con los comandos a usar. Una forma de crearlos es usar Visual Studio Code.

Al abrir Visual Studio Code haga click en Open Folder y seleccione una carpeta donde se guardaran los scripts que haga.



Después haga click derecha en la carpeta, New File y escriba un nombre terminado en .js, a la derecha podrá escribir el código.



Bibliografia

<https://code.visualstudio.com>

<https://www.mongodb.com/es>

<https://docs.mongodb.com/mongodb-shell/>