



mongoDB®





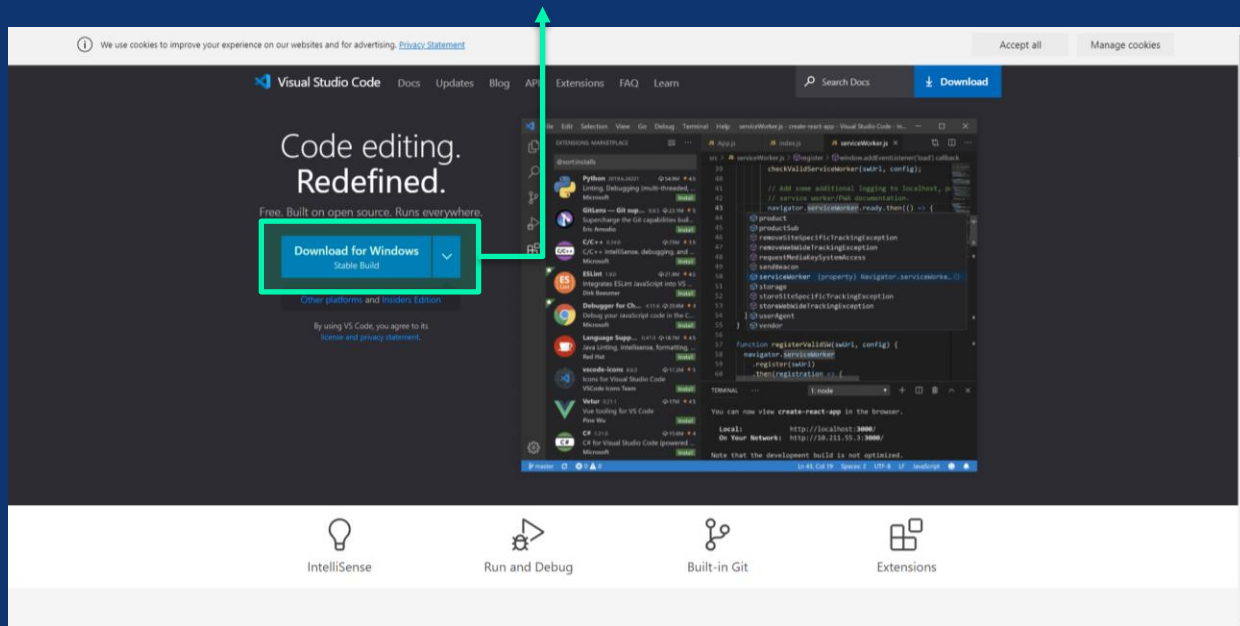
Visual Studio Code



INSTALACIÓN Y CONFIGURACIÓN VSCODE



Accedemos a la página oficial de [VSCode](https://code.visualstudio.com) y hacemos click en la descarga como se muestra en la imagen

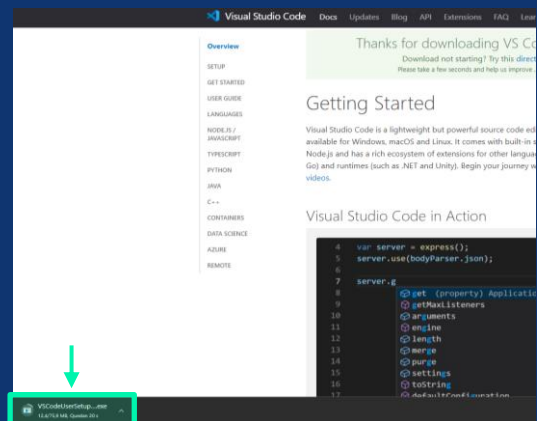
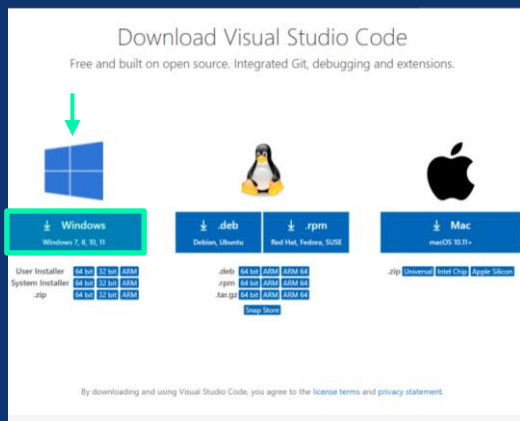


INSTALACIÓN Y CONFIGURACIÓN VSCODE

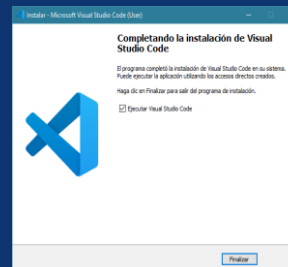
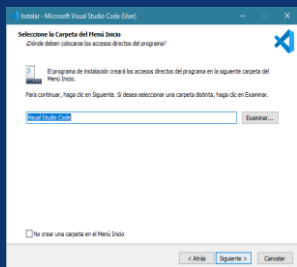
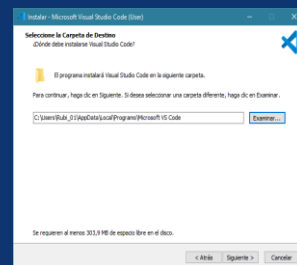
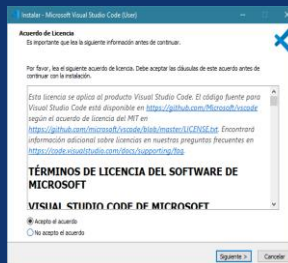
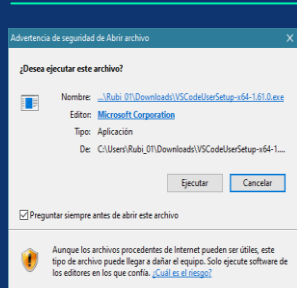


Elegimos la opción dependiendo de nuestro sistema operativo, en mi caso, **Windows 64bits**

En la barra inferior veremos como comienza la descarga



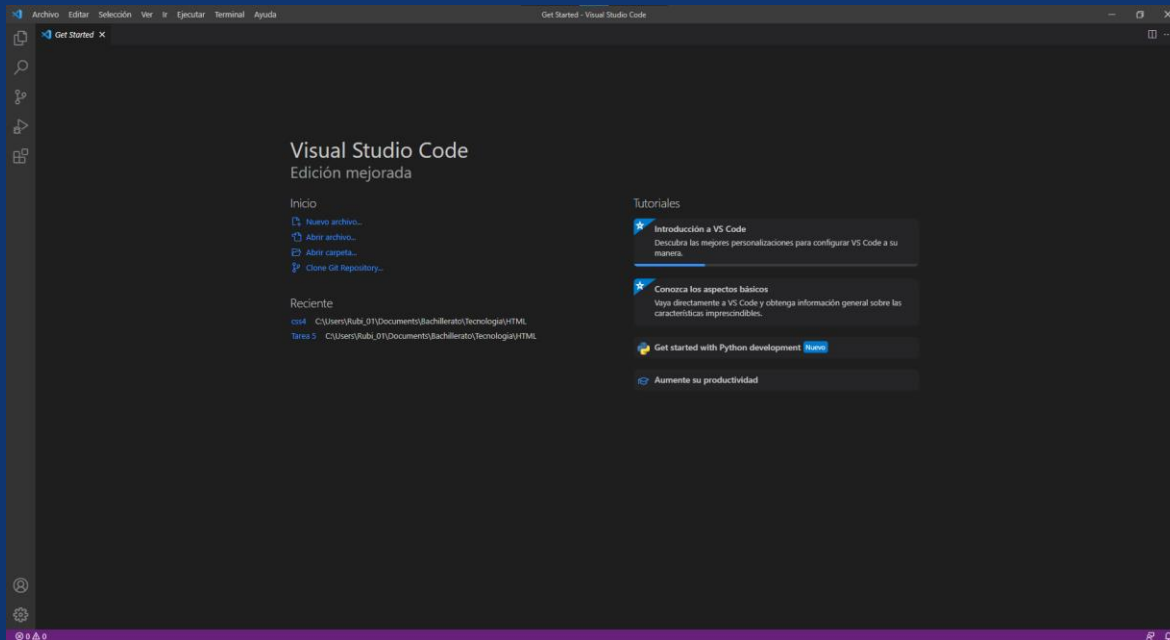
INSTALACIÓN Y CONFIGURACIÓN VSCODE



INSTALACIÓN Y CONFIGURACIÓN VSCODE



Una vez lanzado el programa nos aparecerá una ventana como la que aparece en la imagen

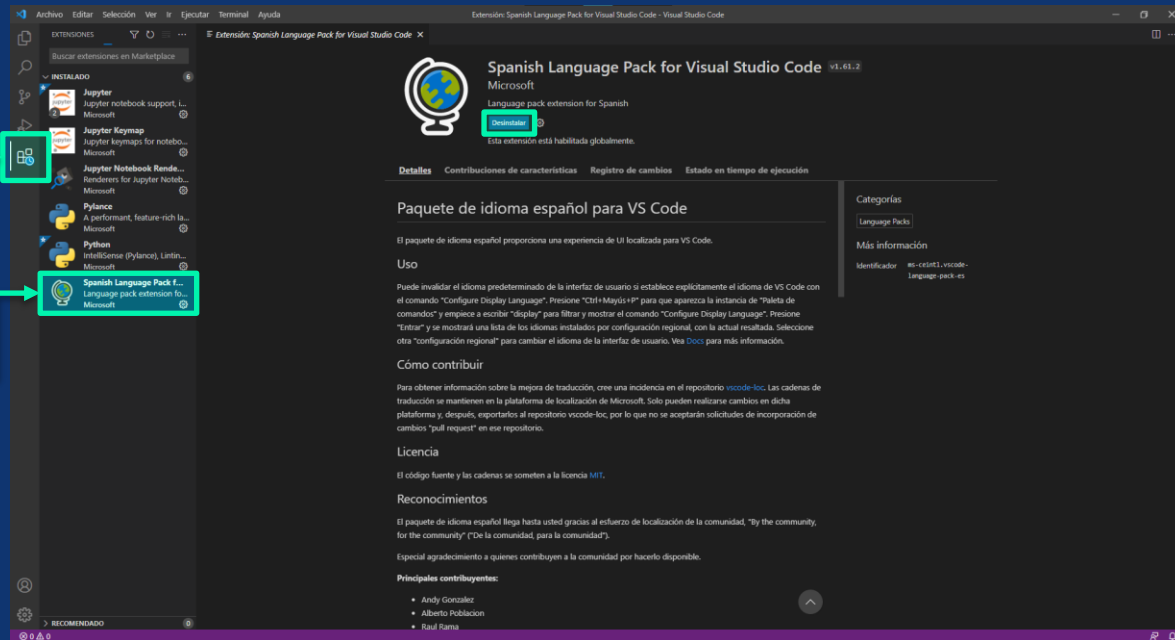


INSTALACIÓN Y CONFIGURACIÓN VSCODE



Una vez instalado accedemos a las **extensiones** en la ultima opción del menú lateral izquierdo

En el apartado de Extensiones buscamos el **paquete de lenguaje Español** y lo instalamos



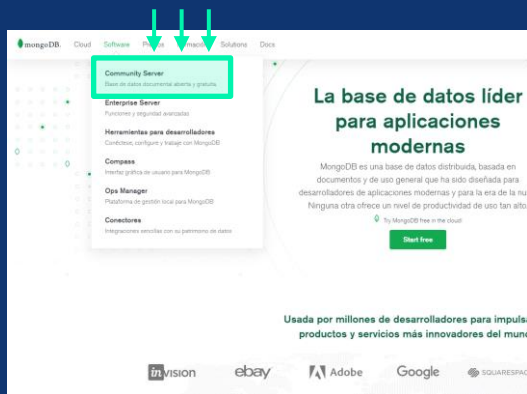


mongoDB®

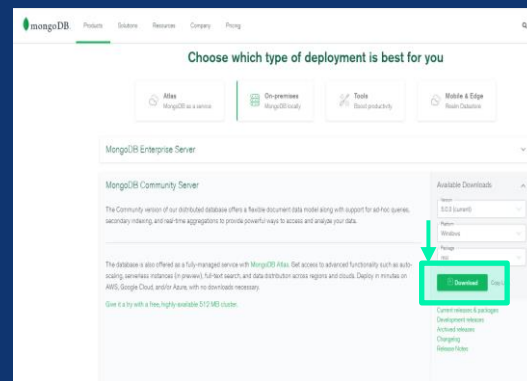


INSTALACIÓN Y CONFIGURACIÓN MONGODB

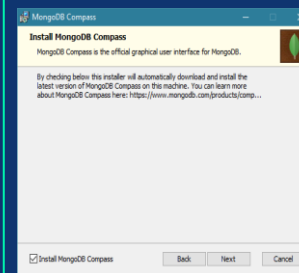
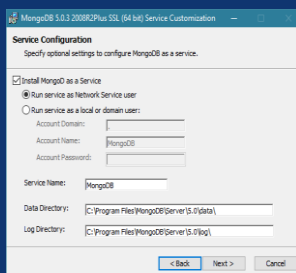
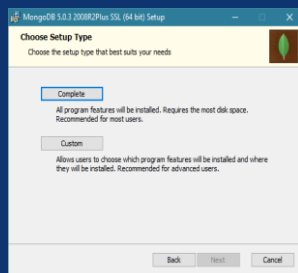
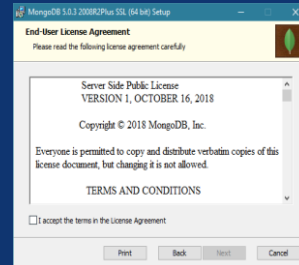
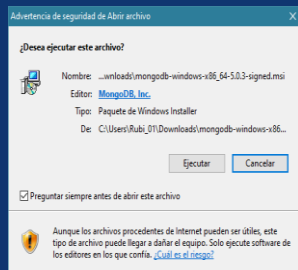
Para instalar [mongoDB](#) entramos en el hipervínculo proporcionado y accedemos a [Software\CommunityServer](#)



Una vez dentro de [Community Server](#) en la parte inferior derecha descargamos la versión que sea compatible con nuestro sistema



INSTALACIÓN Y CONFIGURACIÓN MONGODB

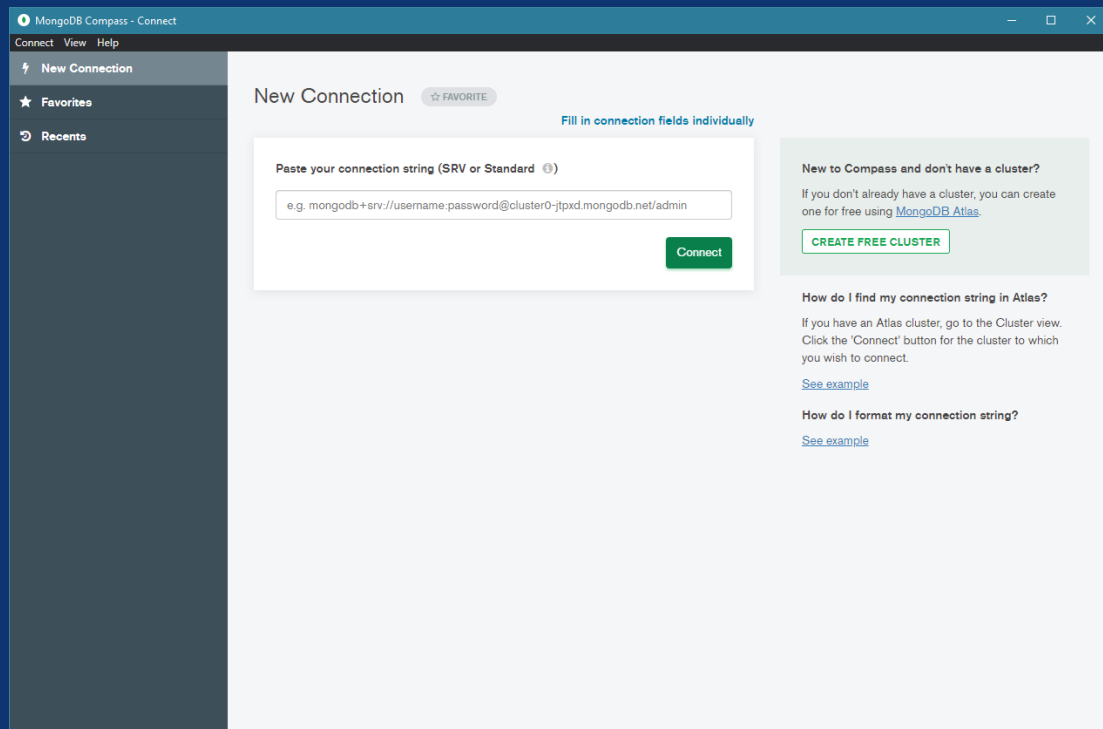


INSTALACIÓN Y CONFIGURACIÓN MONGODB



Si nos fijamos en la instalación, hemos dejado marcada la opción de instalar **MongoDB Compass**.

MongoDB Compass es la interfaz grafica de la base de datos mongo, la cual, por ahora, no utilizaremos

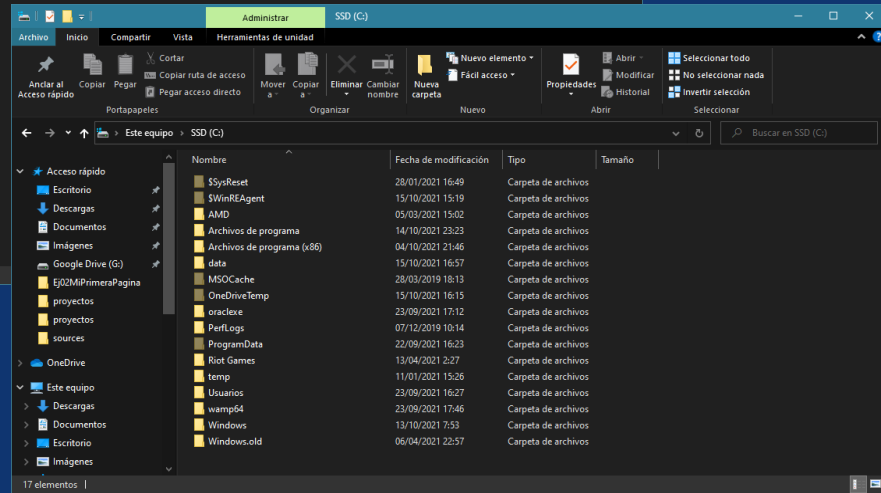
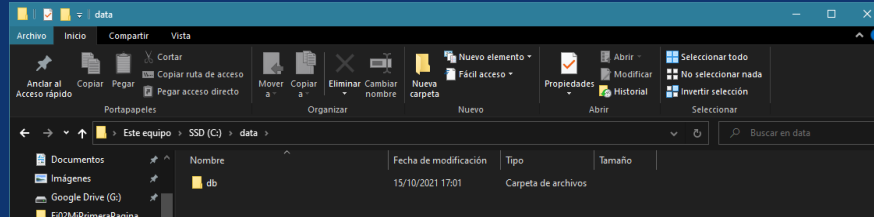


INSTALACIÓN Y CONFIGURACIÓN MONGODB



Para comenzar con la creación de nuestro servidor local de **MongoD** y enlazarlo a la base de datos de **Mongo** debemos:

1. Crear una carpeta llamada **data** en la ruta de nuestro disco (**C:**) y dentro de ella otra carpeta llamada **db**. Aquí se almacenará nuestro servidor local y le permitirá ejecutarse desde la consola



INSTALACIÓN Y CONFIGURACIÓN MONGODB



2. A continuación abrimos la consola de Windows y ejecutaremos el comando **mongod** para iniciar el servidor de mongo

3. Abrimos otra consola donde escribiremos el comando **mongo** para conectarnos al servidor.

Una vez realizado esto, ya tendremos mongo enlazado y listo para ser usado

```
Simbolo del sistema - mongod
able logging settings modifications are required for existing WiredTiger tables,"attr":{"loggingEnabled":true}}
{"t":{"$date":"2021-10-15T17:16:31.813+02:00"},"s":"I", "c":"STORAGE", "id":22262, "ctx":"initandlisten","msg":"Time
stamp monitor starting"}
{"t":{"$date":"2021-10-15T17:16:31.817+02:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Acces
s control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["st
artupWarnings"]}
{"t":{"$date":"2021-10-15T17:16:31.817+02:00"},"s":"W", "c":"CONTROL", "id":22140, "ctx":"initandlisten","msg":"This
server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip
<address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces.
If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["startupWarning
s"]}
{"t":{"$date":"2021-10-15T17:16:31.821+02:00"},"s":"I", "c":"NETWORK", "id":4915702, "ctx":"initandlisten","msg":"Upda
ted wire specification","attr":{"oldSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingI
ternalClient":{"minWireVersion":0,"maxWireVersion":13},"newSpec":{"incomingExternalClient":{"minWireVersion":13,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"newSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"newSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13}}}}}
c:\Users\Rubi_01>mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("1cfff2d5-6b77-4130-9c7f-06eae4089929") }
MongoDB server version: 5.0.3
Control is enabled on this deployment
{"t":{"$date":"2021-10-15T17:16:31.821+02:00"},"s":"W", "c":"CONTROL", "id":22140, "ctx":"initandlisten","msg":"This
server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip
<address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces.
If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["startupWarning
s"]}
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:
2021-10-15T16:27:24.910+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.

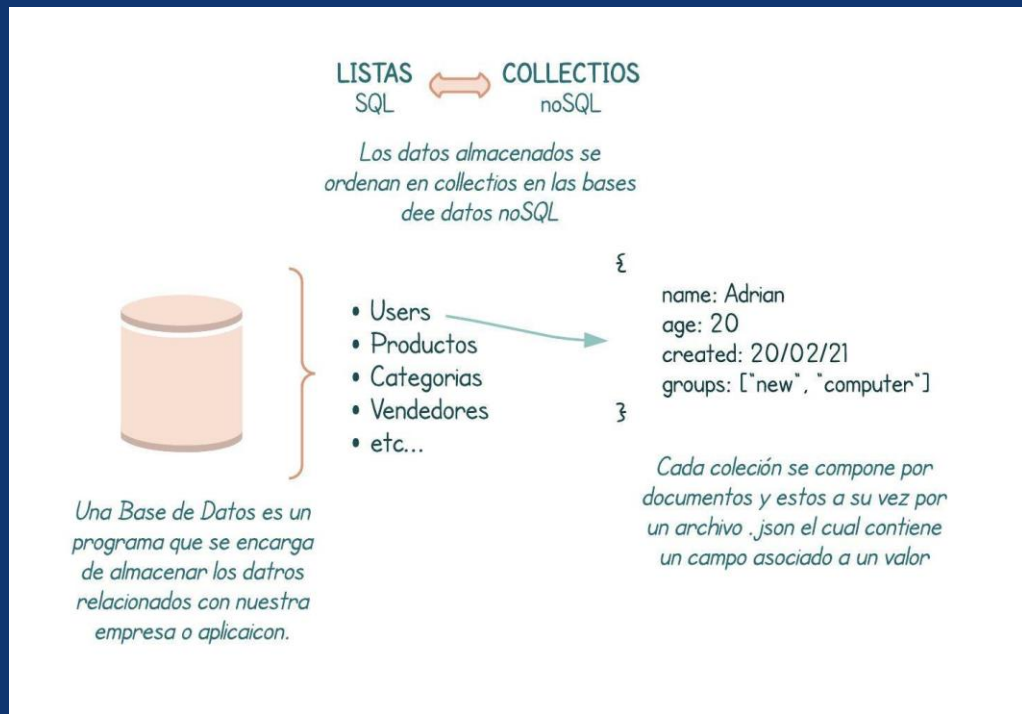
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----
>
```



INSTALACIÓN Y CONFIGURACIÓN MONGODB



Antes de ponernos manos a la obra haremos un breve resumen de como funciona una base de datos **noSQL**



PRIMEROS COMANDOS



Si tenemos una versión igual o superior a la 5.0 no nos hará falta iniciar el servidor con el comando `mongod` ya que solo utilizando `mongo` nos creará y se conectará al servidor en segundo plano.

Con el comando `mongo --version` en la consola de Windows sabremos que versión tenemos instalada

```
Administrador: Símbolo del sistema

C:\WINDOWS\system32>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"
  }
}
```



PRIMEROS COMANDOS



Para interactuar con la base de datos iniciamos la consola e introducimos el comando `mongo` para ejecutar la `mongo Shell`

Básicamente `mongo shell` es un interprete de JavaScript para poder interactuar con la base de datos

Con el comando `cls` limpiaremos la consola

```
Administrador: Símbolo del sistema - mongo
C:\WINDOWS\system32>mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("8c5b4f0a-2654-4ee1-a31e-0f89944ef8b9") }
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:
  2021-10-15T16:27:24.910+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()
---
> cls
```



PRIMEROS COMANDOS



Para comenzar, empezaremos con el comando **db** el cual nos mostrará la base de datos que estamos utilizando. Por defecto mongo nos conecta a una base de datos llamada **test**

Para ver las bases de datos que tenemos disponibles utilizaremos el comando **show dbs**.

Estas bases de datos (admin, config y local) son para el funcionamiento interno de MongoDB, para establecer permisos, usuarios, configurarlos...

Otro comando útil a conocer es **help** el cual nos proporcionará los comandos y palabras reservadas que tiene MongoDB

```
Simbolo del sistema - mongo
> db
test
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> help
db.help()                help on db methods
db.mycoll.help()         help on collection methods
sh.help()                sharding helpers
rs.help()                replica set helpers
help admin              administrative help
help connect            connecting to a db help
help keys               key shortcuts
help misc               misc things to know
help mr                 mapreduce

show dbs                show database names
show collections         show collections in current database
show users              show users in current database
show profile            show most recent system.profile entries with time >= 1ms
show logs               show the accessible logger names
show log [name]         prints out the last segment of log in memory, 'global' is default
use <db name>           set current database
db.mycoll.find()         list objects in collection mycoll
db.mycoll.find( { a : 1 } ) list objects in mycoll where a == 1
it                      result of the last line evaluated; use to further iterate
DBQuery.shellBatchSize = x set default number of items to display on shell
exit                    quit the mongo shell
```



PRIMEROS COMANDOS



Empecemos creando nuestra primera base de datos, para ello introduciremos el comando `use <nombreDB>`, una vez mongoDB se posicionará en ella para empezar a trabajar. Si utilizamos `db` podemos comprobarlo.

Tenemos que tener en cuenta que la base de datos que hemos introducido no se creará hasta que no insertemos como mínimo un dato en ella. Para introducir una colección con un documento en nuestra base de datos utilizaremos el comando `db.<nombreColección>.insert({"name": "laptop"})`

Si introducimos `show dbs` ahora sí nos aparecerá nuestra base de datos. Para ver las colecciones creadas utilizamos `show collections`

```
Simbolo del sistema - mongo
> use webstore
switched to db webstore
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> db.products.insert({"name": "laptop"})
WriteResult({"nInserted" : 1 })
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
webstore 0.000GB
> show collections
products
```



PRIMEROS COMANDOS



Supongamos que una vez creada nuestra base de datos queremos eliminarla, para ello utilizaremos primero el comando `db` para asegurarnos en que base de datos estamos y luego el comando `db.dropDatabase()` y podemos comprobarlo con `show dbs`

```
Simbolo del sistema - mongo
> db
webstore
> db.dropDatabase()
{ "ok" : 1 }
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
>
```



PRIMEROS COMANDOS



Vamos a entrar en detalle en como funcionan las colecciones

Para introducir colecciones de manera rápida y sin necesidad de añadirle documentos en primera instancia, utilizaremos el comando

`db.createCollection("<nombreColección>")`

Si quisiéramos borrar una colección en concreto utilizaríamos

`db.<nombreColección>.drop()`

Simbolo del sistema - mongo

```
> use webstore
switched to db webstore
> db.createCollection("users")
{ "ok" : 1 }
> show collections
users
> db.createCollection("products")
{ "ok" : 1 }
> db.createCollection("sellers")
{ "ok" : 1 }
> show collections
products
sellers
users
> db.products.drop()
true
> show collections
sellers
users
>
```



PRIMEROS COMANDOS



Para profundizar más veamos los **tipos de datos** que podemos introducir en los documentos. Insertamos una colección y dentro de esta un nuevo documento Para comprobar la creación del documento podemos insertar

`db.products.find()`

el cual nos mostrará todos los documentos dentro de esta colección

tips: si después del método `find()` añadimos `.pretty()` nos mostrará el documento ordenado y tabulado

`db.products.find().pretty()`

```
Símbolo del sistema - mongo
> db
webstore
> show collections
> db.products.insert({
...   "name": "laptop",
...   "price": 40.2,
...   "active": false,
...   "created_at": new Date("12/12/1999"),
...   "somedata": [1, "a", []],
...   "maker": {
...     "name": "dell",
...     "version": "xps",
...     "location": {
...       "city": "spain",
...       "address": "etcetc"
...     }
...   }
... })
WriteResult({ "nInserted" : 1 })
> db.products.find()
{ "_id" : ObjectId("6169cb6d52fa49a9e552f67c"), "name" : "laptop", "price" : 40.2, "active" :
false, "created_at" : ISODate("1999-12-11T23:00:00Z"), "somedata" : [ 1, "a", [ ] ], "maker"
: { "name" : "dell", "version" : "xps", "location" : { "city" : "spain", "address" : "etcetc"
} } }
```

String

Number

Boolean

Objeto de JavaScript

Array

Documento, es decir un objeto
Json puede contener otro objeto
sin limite de reiteración



PRIMEROS COMANDOS



Imaginemos que tenemos una colección con varios documentos y sus propiedades y queremos hacer una búsqueda concreta con un parámetro definido, por ejemplo, dentro de `products` queremos buscar solo los productos que tengan como nombre mouse, para ello utilizaremos el método `find()` introduciendo los parámetros de búsqueda:

```
db.products.find({"name": "mouse"})
```

Los parámetros de búsqueda se pueden concatenar en nuestro método `find()` para ser más precisos

```
db.products.find({"tag": "computer", "name": "monitor"})
```

```
> db.products.find()
{ "_id" : ObjectId("616a24ad8b255eb208f99b41"), "name" : "mouse", "descriptions" : "razer mouse", "tags" : [ "computers", "gaming" ], "quantity" : 14, "created_at" : ISO
Date("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b42"), "name" : "monitor", "descriptions" : "lg monitor", "tags" : [ "computers", "gaming" ], "quantity" : 3, "created_at" : ISO
Date("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b43"), "name" : "keyboard", "descriptions" : "corsair keyboard", "tags" : [ "computers", "gaming" ], "quantity" : 6, "created_at
" : ISODate("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b44"), "name" : "laptop", "descriptions" : "samsung laptop", "tags" : [ "computers", "gaming" ], "quantity" : 15, "created_at" :
ISODate("2021-10-16T01:02:37.204Z") }
> db.products.find({"name": "mouse"})
{ "_id" : ObjectId("616a24ad8b255eb208f99b41"), "name" : "mouse", "descriptions" : "razer mouse", "tags" : [ "computers", "gaming" ], "quantity" : 14, "created_at" : ISO
Date("2021-10-16T01:02:37.204Z") }
> db.products.find({"tags": "computers", "name": "monitor"})
{ "_id" : ObjectId("616a24ad8b255eb208f99b42"), "name" : "monitor", "descriptions" : "lg monitor", "tags" : [ "computers", "gaming" ], "quantity" : 3, "created_at" : ISO
Date("2021-10-16T01:02:37.204Z") }
>
```



PRIMEROS COMANDOS



Si nos fijamos en la diapositiva anterior, cuando realizamos una búsqueda de un documento, mongo nos devuelve el documento que cumple con los parámetros de búsqueda y todos sus datos. Si solo queremos que nos devuelva datos concretos y no su totalidad, será tan fácil como marcar esos datos :1 para visualizarlos y :0 para no, veamos un ejemplo:

```
db.products.find({"tags": "computers"}, {"name": 1, "description": 1, "_id": 0})
```

```
> db.products.find({"tags": "computers"})
{ "_id" : ObjectId("616a24ad8b255eb208f99b41"), "name" : "mouse", "descriptions" : "razer mouse", "tags" : [ "computers", "gaming" ], "quantity" : 14, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b42"), "name" : "monitor", "descriptions" : "lg monitor", "tags" : [ "computers", "gaming" ], "quantity" : 3, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b43"), "name" : "keyboard", "descriptions" : "corsair keyboard", "tags" : [ "computers", "gaming" ], "quantity" : 6, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b44"), "name" : "laptop", "descriptions" : "samsung laptop", "tags" : [ "computers", "gaming" ], "quantity" : 15, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }

> db.products.find({"tags": "computers"}, {"name": 1, "descriptions": 1})
{ "_id" : ObjectId("616a24ad8b255eb208f99b41"), "name" : "mouse", "descriptions" : "razer mouse" }
{ "_id" : ObjectId("616a24ad8b255eb208f99b42"), "name" : "monitor", "descriptions" : "lg monitor" }
{ "_id" : ObjectId("616a24ad8b255eb208f99b43"), "name" : "keyboard", "descriptions" : "corsair keyboard" }
{ "_id" : ObjectId("616a24ad8b255eb208f99b44"), "name" : "laptop", "descriptions" : "samsung laptop" }
```



TIPS DENTRO DE LAS BÚSQUEDAS



`db.products.find({"tags":"computers"}).sort({name:1})` -> ordena la búsqueda por orden alfabético.
`db.products.find({"tags":"computers"}).limit(2)` -> te muestra los dos últimos documentos que concuerden.
`db.products.find({"tags":"computers"}).count()` -> muestra la cantidad de documentos dentro de la colección.

En mongoDB se pueden utilizar funciones factoriales propias de JavaScript y palabras reservadas de mongoDB

`db.products.find().forEach(product => print("Product Price: "+ product.price))` -> en este caso, muestra indefinido porque no hay atributos de precio.

```
> db.products.find({"tags":"computers"}).sort({name:1})
{ "_id" : ObjectId("616a24ad8b255eb208f99b43"), "name" : "keyboard", "descriptions" : "corsair keyboard", "tags" : [ "computers", "gaming" ], "quantity" : 6, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b44"), "name" : "laptop", "descriptions" : "samsung laptop", "tags" : [ "computers", "gaming" ], "quantity" : 15, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b42"), "name" : "monitor", "descriptions" : "lg monitor", "tags" : [ "computers", "gaming" ], "quantity" : 3, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b41"), "name" : "mouse", "descriptions" : "razer mouse", "tags" : [ "computers", "gaming" ], "quantity" : 14, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }
> db.products.find({"tags":"computers"}).limit(2)
{ "_id" : ObjectId("616a24ad8b255eb208f99b41"), "name" : "mouse", "descriptions" : "razer mouse", "tags" : [ "computers", "gaming" ], "quantity" : 14, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }
{ "_id" : ObjectId("616a24ad8b255eb208f99b42"), "name" : "monitor", "descriptions" : "lg monitor", "tags" : [ "computers", "gaming" ], "quantity" : 3, "created_at" : ISODate("2021-10-16T01:02:37.204Z") }
> db.products.find({"tags":"computers"}).count()
4
> db.products.find().forEach(product => print("Product Price: "+ product.price))
Product Price: undefined
Product Price: undefined
Product Price: undefined
Product Price: undefined
>
```



UPDATE



Como observamos en la diapositiva anterior, nuestros documentos no tienen el campo de precio, imaginemos que necesitamos añadir el precio de nuestros articulo en la base de datos, para ello utilizaremos el método *update* de la siguiente forma:

```
db.products.update({"name":"laptop"},  
{"name":"laptop", "price":99.99})
```

Hay que tener en cuenta que el método *update* reemplaza todo el documento que coincida con el primer objeto introducido (name:laptop) por lo introducido en el segundo objeto (name: laptop, price: 99.99)

```
> db.products.update({"name":"laptop"}, {"name":"laptop", "price":99.99})  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.products.find().pretty()  
{  
  "_id" : ObjectId("616c74e4c8faa2e6a5ffbcb6"),  
  "name" : "mouse",  
  "descriptions" : "razer mouse",  
  "price" : 34.05,  
  "tags" : [  
    "computers",  
    "gaming"  
  ],  
  "quantity" : 14,  
  "created_at" : ISODate("2021-10-17T19:09:24.372Z")  
}  
{  
  "_id" : ObjectId("616c74e4c8faa2e6a5ffbcb7"),  
  "name" : "monitor",  
  "descriptions" : "lg monitor",  
  "price" : 135.99,  
  "tags" : [  
    "computers",  
    "gaming"  
  ],  
  "quantity" : 3,  
  "created_at" : ISODate("2021-10-17T19:09:24.372Z")  
}  
{  
  "_id" : ObjectId("616c74e4c8faa2e6a5ffbcb8"),  
  "name" : "keyboard",  
  "descriptions" : "corsair keyboard",  
  "price" : 99.99,  
  "tags" : [  
    "computers",  
    "gaming"  
  ],  
  "quantity" : 6,  
  "created_at" : ISODate("2021-10-17T19:09:24.372Z")  
}  
  
{"_id" : ObjectId("616c74e4c8faa2e6a5ffbcb9"),  
 "name" : "laptop",  
 "price" : 99.99  
}
```



TIPS DENTRO DE UPDATE



Cuando tenemos documentos muy extensos dentro de nuestra base de datos no queremos actualizarlo y tener que reescribir la información que no queremos cambiar, para ello utilizaremos el operador de actualización **\$set** junto con el método **update** de la siguiente forma:

```
db.products.update({"name":"laptop"}, {$set:{"stock":"false"}})
```

Si utilizamos el método **update** para actualizar un campo del documento pero este no existe, podemos hacer que lo cree con el operador **upsert: true**

El método **update** se puede utilizar con el operador **\$inc** para incrementar el valor de un number dentro de nuestros documentos y con el método **\$rename**

```
> db.products.find()
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "mouse", "descriptions" : "razer mouse", "price" : 34.05, "tags" : [ "computers", "gaming" ], "quantity" : 14, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "monitor", "descriptions" : "lg monitor", "price" : 135.99, "tags" : [ "computers", "gaming" ], "quantity" : 3, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "keyboard", "descriptions" : "corsair keyboard", "price" : 99.99, "tags" : [ "computers", "gaming" ], "quantity" : 6, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "laptop", "descriptions" : "samsung laptop", "price" : 120, "tags" : [ "computers", "gaming" ], "quantity" : 15, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }

> db.products.update({"name":"laptop"},{$set:{"stock":"false"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.products.find()
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "mouse", "descriptions" : "razer mouse", "price" : 34.05, "tags" : [ "computers", "gaming" ], "quantity" : 14, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "monitor", "descriptions" : "lg monitor", "price" : 135.99, "tags" : [ "computers", "gaming" ], "quantity" : 3, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "keyboard", "descriptions" : "corsair keyboard", "price" : 99.99, "tags" : [ "computers", "gaming" ], "quantity" : 6, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "laptop", "descriptions" : "samsung laptop", "price" : 120, "tags" : [ "computers", "gaming" ], "quantity" : 15, "created_at" : ISODate("2021-10-17T18:46:36.941Z"), "stock" : "false" }

> db.products.update({"name":"destok"},{$set:{"descriptions":"gaming destok"}},{upsert:true})
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 1,
  "nModified" : 0,
  "_id" : ObjectId("616c721ca0450e54feb576b6")
})
> db.products.find()
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "mouse", "descriptions" : "razer mouse", "price" : 34.05, "tags" : [ "computers", "gaming" ], "quantity" : 14, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "monitor", "descriptions" : "lg monitor", "price" : 135.99, "tags" : [ "computers", "gaming" ], "quantity" : 3, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "keyboard", "descriptions" : "corsair keyboard", "price" : 99.99, "tags" : [ "computers", "gaming" ], "quantity" : 6, "created_at" : ISODate("2021-10-17T18:46:36.941Z") }
{ "_id" : ObjectId("616c6f8cc8faa2e6a5ffbce"), "name" : "laptop", "descriptions" : "samsung laptop", "price" : 120, "tags" : [ "computers", "gaming" ], "quantity" : 15, "created_at" : ISODate("2021-10-17T18:46:36.941Z"), "stock" : "false" }
{ "_id" : ObjectId("616c721ca0450e54feb576b6"), "name" : "destok", "descriptions" : "gaming destok" }

>
> db.products.update({"name":"laptop"}, {$inc:{"price":0.01}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.products.find({"name":"laptop"})
{ "_id" : ObjectId("616c74e4c8faa2e6a5ffbce"), "name" : "laptop", "price" : 120.01 }
```

OPERADORES-COMPARACIÓN



<u>Nombre</u>	<u>Descripción</u>
\$eq	Coincide con valores que son iguales a un valor especificado.
\$gt	Coincide con valores que son mayores que un valor especificado.
\$gte	Coincide con valores que son mayores o iguales a un valor especificado.
\$in	Coincide con cualquiera de los valores especificados en una matriz.
\$lt	Coincide con valores que son menores que un valor especificado.
\$lte	Coincide con valores que son menores o iguales a un valor especificado.
\$ne	Coincide con todos los valores que no son iguales a un valor especificado.
\$nin	No coincide con ninguno de los valores especificados en una matriz.



OPERADORES-LÓGICOS



<u>Nombre</u>	<u>Descripción</u>
\$and	Coincide con valores que son iguales a un valor especificado.
\$not	Coincide con valores que son mayores que un valor especificado.
\$nor	Coincide con valores que son mayores o iguales a un valor especificado.
\$or	Coincide con cualquiera de los valores especificados en una matriz.





mongoDB®

Ahora tenemos unos conocimientos básicos de como funciona mongoDB. Facilito un [link](#) con un chuleterio de código básico y en la carpeta src se encuentran los archivos de búsqueda, inserción, actualizados, etc.

