

A dark blue vertical bar runs along the left edge of the slide. A blue arrow-shaped banner points to the right from this bar, containing the date. In the bottom-left corner, there are several thin, curved lines in shades of blue and grey, resembling stylized grass or reeds.

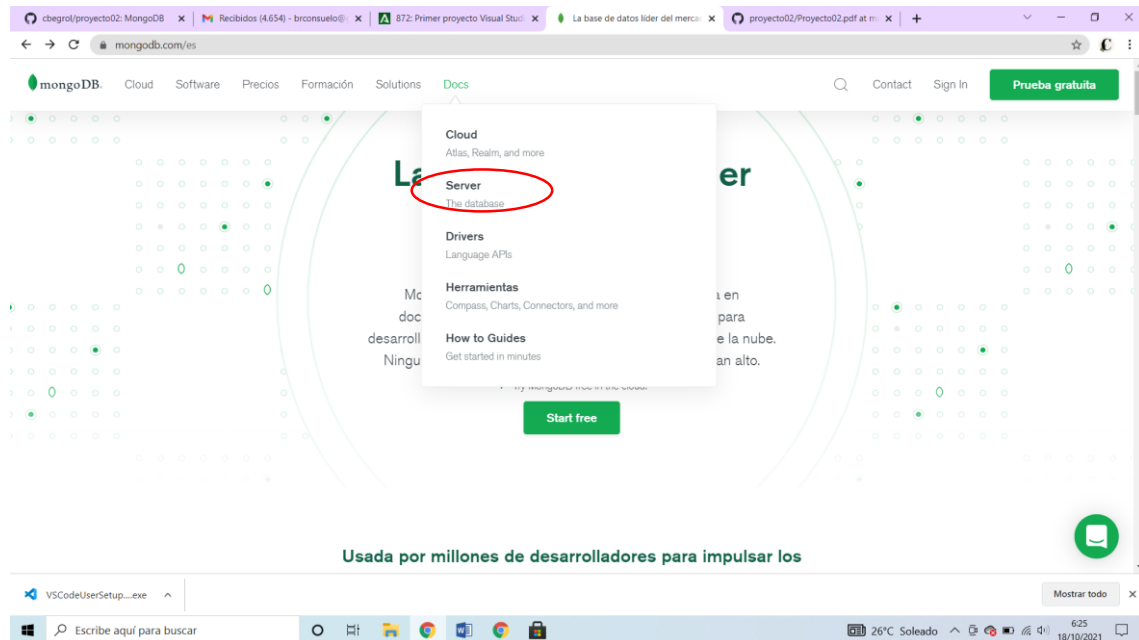
18-10-2021

PRIMEROS COMANDOS EN MONGO SHELL Y MÉTODOS

www.mongodb.com

Consuelo Begines Roldán
1º DAM GESTIÓN BASE DE DATOS

En la web de Mongo, entramos en el apartado “Docs” → “Server”, y aparece el manual de mongo



Si nos vamos a PowerShell, escribimos mongo –versión, nos situamos en nuestra carpeta src y volvemos a escribir por segunda vez mongo, vemos como ya nos aparece mongo Shell

```
Windows PowerShell
Warning: the "mongo" shell has been superseded by "mongosh",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:
  2021-10-18T02:09:28.999+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
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\brcon> 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\brcon> cd C:\Users\brcon\OneDrive\Escritorio\1ºDAM ASIGNATURAS\GBD\ADOLFO\proyecto02\src
PS C:\Users\brcon\OneDrive\Escritorio\1ºDAM ASIGNATURAS\GBD\ADOLFO\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("244a7a50-8d46-4f59-9220-9c90ae712382") }
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:
2021-10-18T02:09:28.999+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
```

Primero nos muestra la versión que estamos usando “mongo –version”

Después, como ya hemos dicho, nos situamos en el directorio “src”

Y usaremos el comando “cls”, para limpiar la consola.

Con esto, ya estaríamos en mongoshell.

```
Windows PowerShell

> 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

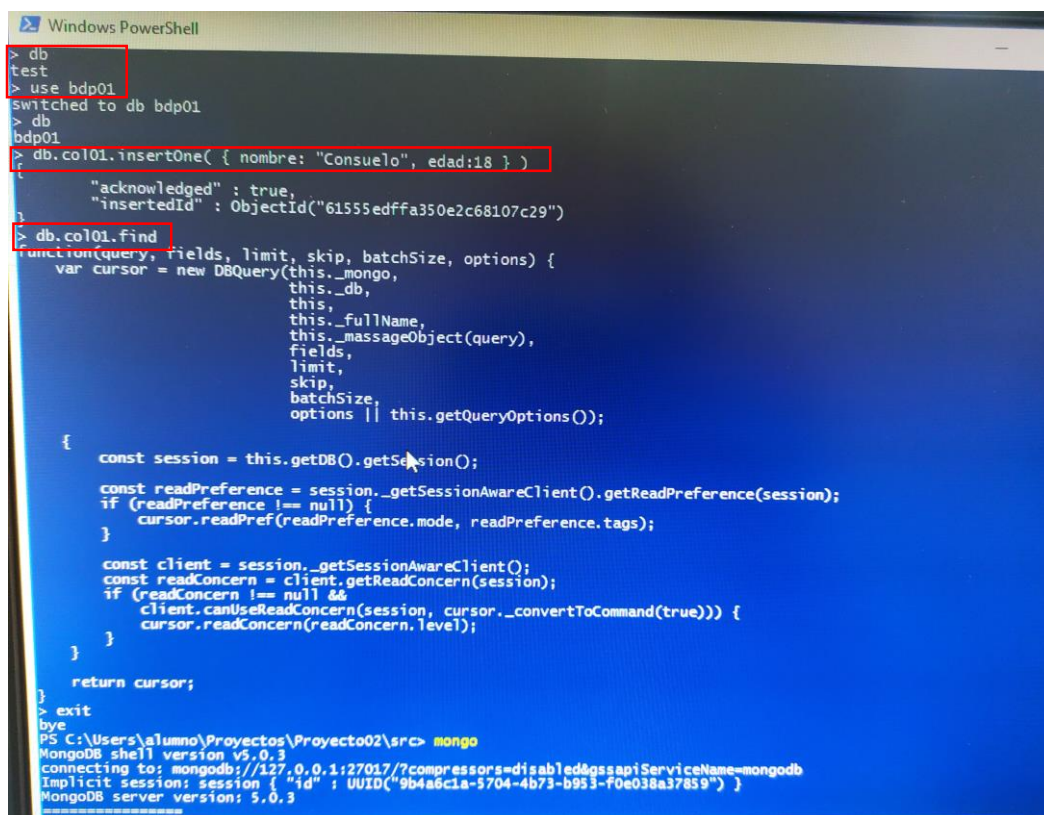
show dbs                 show database names
show collections         show collections in current database
show users              show users in current database
show profile            show most recent system.profile entries with time >= 1ms
show logs               show the accessible logger names
show 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
```

La información del servidor se organiza en Base de Datos. Al poner >db, nos ubicamos en la base de datos que estamos utilizando. Por defecto, nos aparece la base de datos "text".

Una base de datos se organiza en conexiones que a su vez se organizan en documentos tipo .json que está entre corchetes {} y tiene campo y valor.

Para poder comprobar las bases de datos que tenemos disponibles, metemos el comando "show dbs"

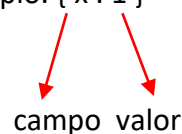
El siguiente comando seria "help", el cual nos mostrará los comando sy palabras que tiene mongoDB.



```
Windows PowerShell
> db
test
> use bdp01
switched to db bdp01
> db
bdp01
> db.col01.insertOne( { nombre: "Consuelo", edad:18 } )
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61555edffa350e2c68107c29")
}
> db.col01.find
function(query, fields, limit, skip, batchSize, options) {
  var cursor = new DBQuery(this._mongo,
    this._db,
    this._fullName,
    this._messageObject(query),
    fields,
    limit,
    skip,
    batchSize,
    options || this.getQueryOptions());
  {
    const session = this.getDB().getSession();
    const readPreference = session._getSessionAwareClient().getReadPreference(session);
    if (readPreference !== null) {
      cursor.readPref(readPreference.mode, readPreference.tags);
    }
    const client = session._getSessionAwareClient();
    const readConcern = client.getReadConcern(session);
    if (readConcern !== null &&
      client.canUseReadConcern(session, cursor._convertToCommand(true))) {
      cursor.readConcern(readConcern.level);
    }
  }
  return cursor;
}
> exit
bye
PS C:\Users\alumno\Proyectos\Proyecto02\src> mongo
MongoDB shell version 5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("9b4a6c1a-3704-4b73-b953-f0e038a37859") }
MongoDB server version: 5.0.3
```

Nos ubicamos en la base de datos y nos dice que cual queremos usar "bdp01"

Insertamos un valor entre llaves, ahí aparece un ejemplo. { x : 1 }

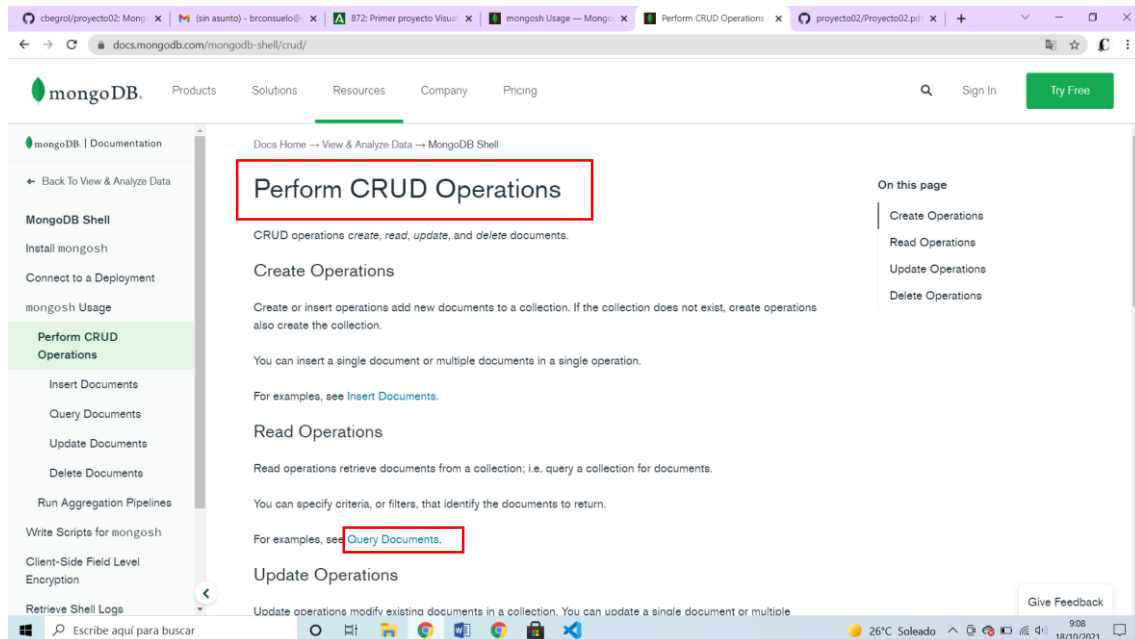


Decimos que instale el documento en la colección llamada myCollection de la Base de Datos mynewdatabase.

>db.col01.insertOne(), metiendo entre paréntesis el método que queremos insertar.

Para consultar el documento, insertamos `> db.col01.find` que nos mostrará los documentos que tenemos dentro de la colección.

Podemos ver la sintaxis de este comando en la página de mongodb, entrando en Perform CRUD Operations y seguidamente en “consultar documentos”, donde encontraremos el método `find()`:



Dentro de PowerShell, introducimos “`>db.col01.find`”, seguidamente “`help`” mediante el cual aparecerán los comando, “`show dbs`” y “`show collections`”:

```
Windows PowerShell
> db.col01.find()
uncaught exception: TypeError: db.col01.find is not a function :
a(shell):1:1
> 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

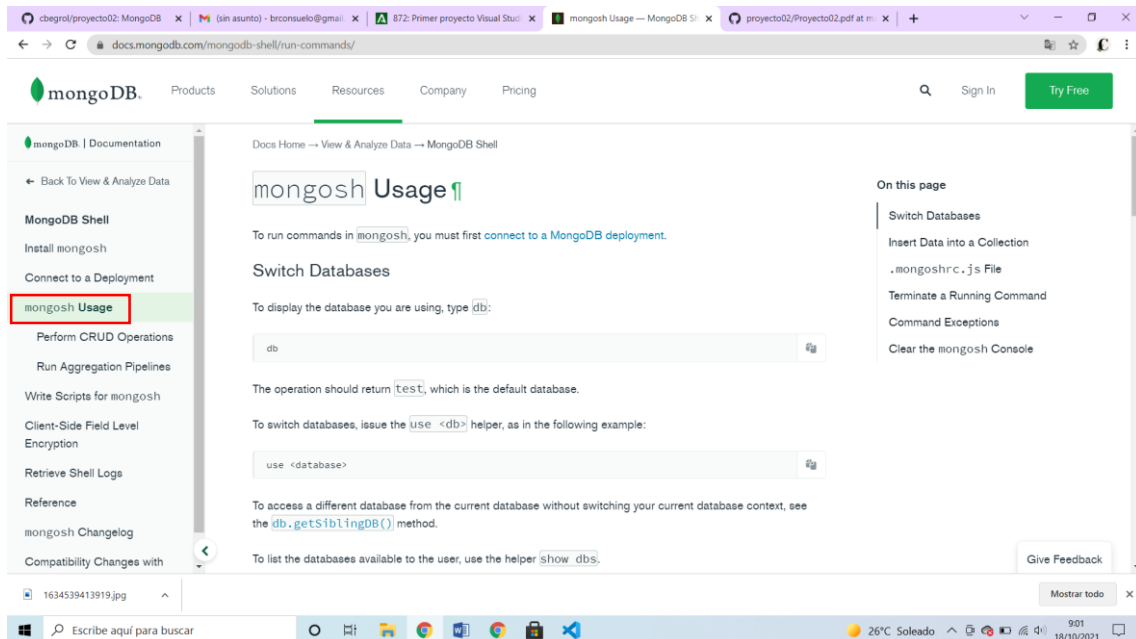
> show dbs
admin 0.000GB
bdp01 0.000GB
config 0.000GB
local 0.000GB
> use bdp01
switched to db bdp01
> show collections
col01
>
```

Le volvemos a pedir que nos muestre las colecciones “show collections” y finalmente añadimos “>db.col01.find()”.

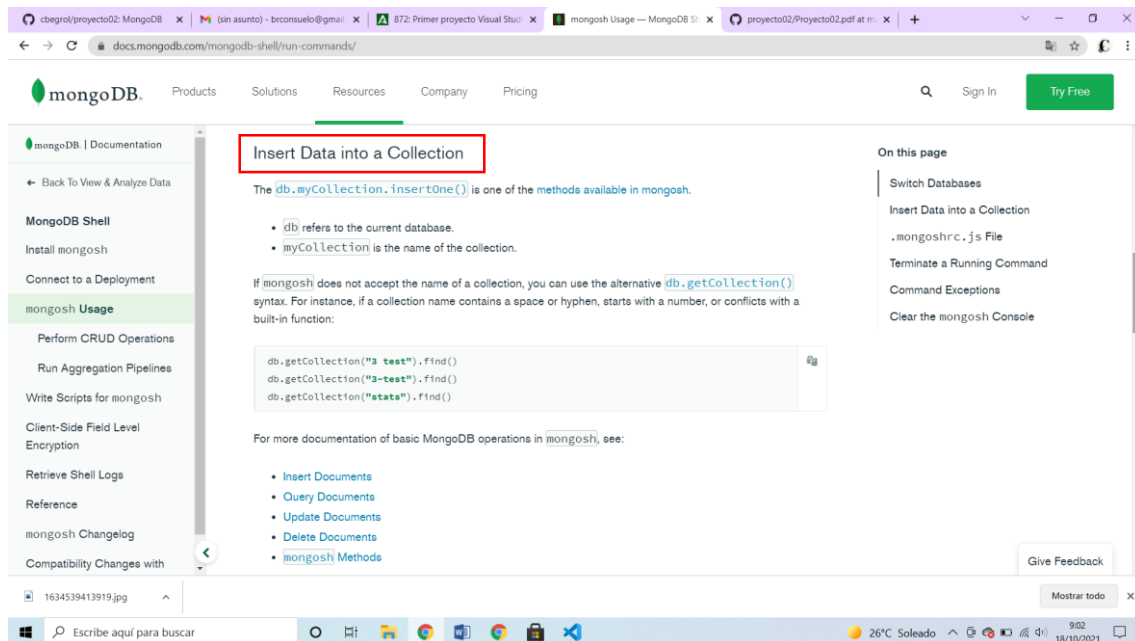
Aparece por defecto “-id” que es el identificador único que asocia mongodb al documento.

```
> show collections
col01
db.col01.find()
{"_id" : ObjectId("61555edffa350e2c68107c29"), "nombre" : "Consuelo", "edad" : 18 }
```

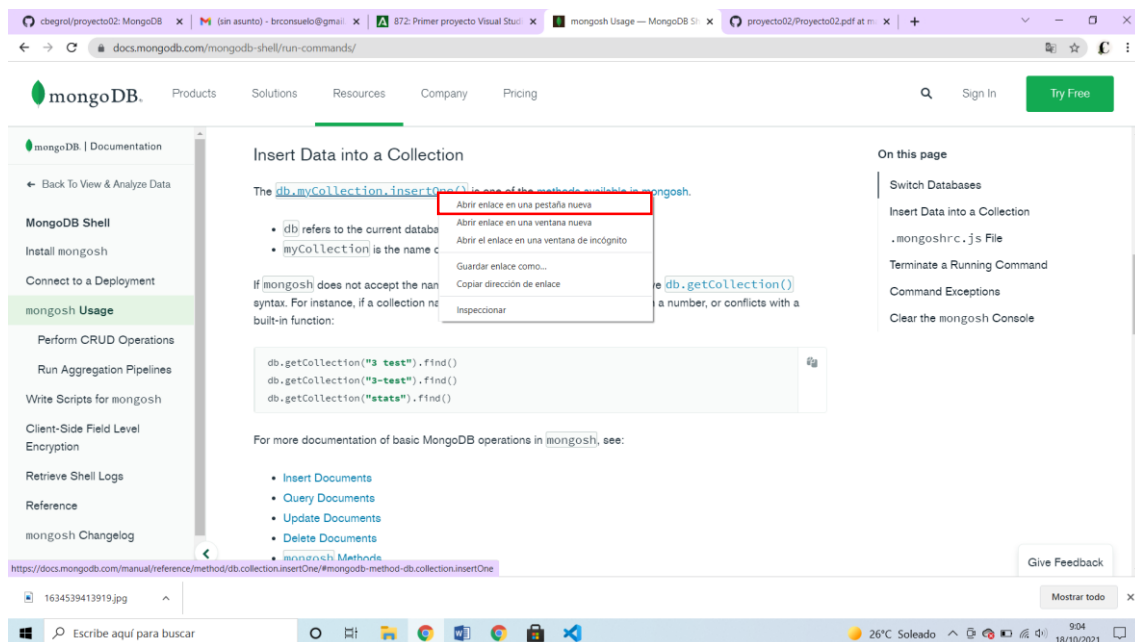
Para abrir la sintaxis de mongosh, entramos en mongosh usage:



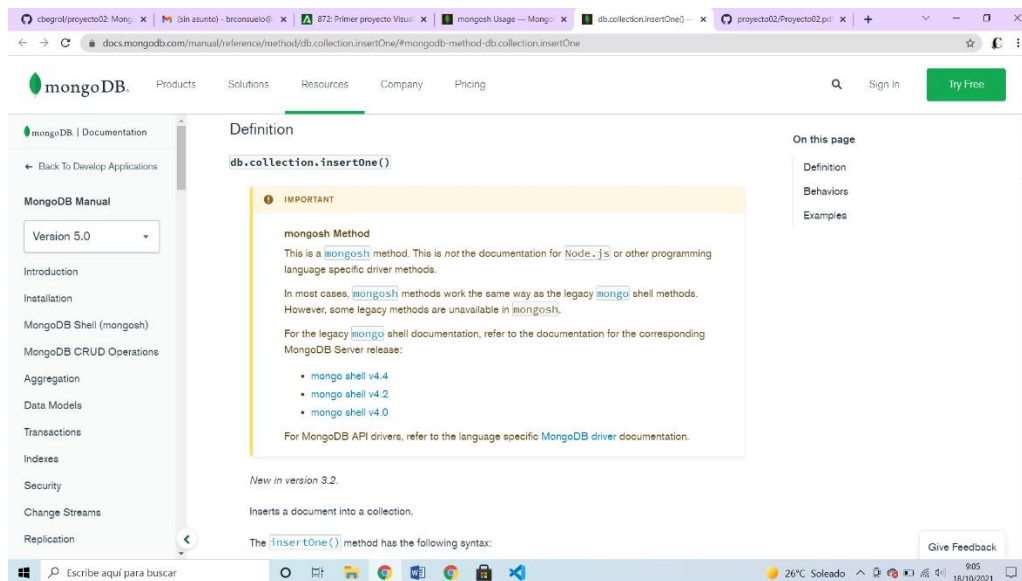
Apartado de insertar datos en una colección:



Abrimos el enlace en una pestaña nueva:



Aquí aparece la sintaxis del sistema:



Se pueden pasar 2 parámetros a insertOne, uno es un documento y el otro es opcional

Operadores de consulta y proyección

Comparación

Nombre	Descripción
\$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.

Nombre	Descripción
--------	-------------

`$nin`

No coincide con ninguno de los valores especificados en una matriz.

Lógico

Nombre	Descripción
--------	-------------

`$and`

Una cláusula de consulta con una lógica `AND` devuelve todos los documentos que cumplen las condiciones de ambas cláusulas.

`$not`

Invierte el efecto de una expresión de consulta y devuelve documentos que *no* coinciden con la expresión de consulta.

`$nor`

Una cláusula de consulta con una lógica `NOR` devuelve todos los documentos que no coinciden con ambas cláusulas.

`$or`

Una cláusula de consulta con `OR` devuelve lógicas de todos los documentos que cumplen las condiciones de cualquiera de las cláusulas.

Elemento

Nombre	Descripción
--------	-------------

`$exists`

Coincide con documentos que tienen el campo especificado.

`$type`

Selecciona documentos si un campo es del tipo especificado.

Evaluación

Nombre	Descripción
--------	-------------

`$expr`

Permite el uso de expresiones de agregación dentro del lenguaje de consulta.

`$jsonSchema`

Validar documentos con el esquema JSON proporcionado.

<code>\$mod</code>	Realiza una operación de módulo sobre el valor de un campo y selecciona documentos con un resultado específico.
<code>\$regex</code>	Selecciona documentos donde los valores coinciden con una expresión regular especificada.
<code>\$text</code>	Realiza búsqueda de texto.
<code>\$where</code>	Coincide con documentos que satisfacen una expresión de JavaScript.

Geoespacial

Nombre	Descripción
<code>\$geoIntersects</code>	Selecciona geometrías que se cruzan con una geometría GeoJSON . El índice 2dsphere admite archivos <code>\$geoIntersects</code> .
<code>\$geoWithin</code>	Selecciona geometrías dentro de una geometría GeoJSON delimitante . Los índices 2dsphere y 2d son compatibles <code>\$geoWithin</code> .
<code>\$near</code>	Devuelve objetos geoespaciales próximos a un punto. Requiere un índice geoespacial. Los índices 2dsphere y 2d son compatibles <code>\$near</code> .
<code>\$nearSphere</code>	Devuelve objetos geoespaciales próximos a un punto de una esfera. Requiere un índice geoespacial. Los índices 2dsphere y 2d son compatibles <code>\$nearSphere</code> .

Matriz

Nombre	Descripción
<code>\$all</code>	Coincide con matrices que contienen todos los elementos especificados en la consulta.
<code>\$elemMatch</code>	Selecciona documentos si el elemento del campo de matriz coincide con todas las <code>\$elemMatch</code> condiciones especificadas .
<code>\$size</code>	Selecciona documentos si el campo de matriz tiene un tamaño especificado.

Bit a bit

Nombre	Descripción
<code>\$bitsAllClear</code>	Coincide con valores numéricos o binarios en el que un conjunto de posiciones de bit <i>todo</i> tienen un valor de <code>0</code> .
<code>\$bitsAllSet</code>	Coincide con valores numéricos o binarios en el que un conjunto de posiciones de bit <i>todo</i> tienen un valor de <code>1</code> .
<code>\$bitsAnyClear</code>	Coincide con valores numéricos o binarios en los que <i>cualquier</i> bit de un conjunto de bit tiene un valor de <code>0</code> .
<code>\$bitsAnySet</code>	Coincide con valores numéricos o binarios en los que <i>cualquier</i> bit de un conjunto de bit tiene un valor de <code>1</code> .

Operadores de proyección

Nombre	Descripción
<code>\$</code>	Proyecta el primer elemento de una matriz que coincide con la condición de la consulta.
<code>\$elemMatch</code>	Proyecta el primer elemento de una matriz que coincide con la <code>\$elemMatch</code> condición especificada .
<code>\$meta</code>	Proyecta la puntuación del documento asignada durante la <code>\$text</code> operación.
<code>\$slice</code>	Limita el número de elementos proyectados de una matriz. Admite omitir y limitar cortes.

Operadores varios

Nombre	Descripción
<code>\$comment</code>	Agrega un comentario a un predicado de consulta.
<code>\$rand</code>	Genera un flotador aleatorio entre 0 y 1.