Contenido

[**1-** **MongoDB** 2](#_Toc126254252)

[1.1- Instalación MongoDB 2](#_Toc126254253)

[1.1.1- Usar solo consola de mongo 2](#_Toc126254254)

[1.1.2- Usar mongo con interfaz gráfica (RECOMENDADO) 3](#_Toc126254255)

[**2-** **NodeJS** 5](#_Toc126254256)

[**3-** **Express** 6](#_Toc126254257)

# **MongoDB**

Son BBDD no relacionales.  
No usamos sql, sino que usamos JavaScript.  
No hay tablas, hay colecciones (json, bson).

## Instalación MongoDB

Ir a la web de MongoDB y descargar Community Server  
<https://www.mongodb.com/try/download/community>

Elegir instalación **completa**.  
Tener marcadas las opciones de instalar como **Servicio** e instalar **MongoDB Compass**.

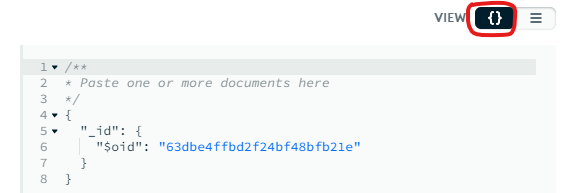
### Usar mongoDB con interfaz gráfica (RECOMENDADO)

Abrir el **MongoDB Compass**

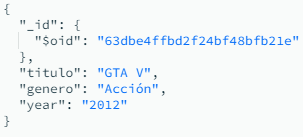
Con la interfaz gráfica puedo crear BBDD, Collections e insertar registros de manera intuitiva.

Para añadir un registro, le doy a ADD DATA > Insert document

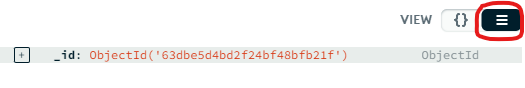
Forma 1:



Y le pongo los datos después del \_id de esta forma.



Forma 2:



Le voy dando al + y añado los campos de forma visual.



### Usar solo consola de mongoDB

1. Manualmente, crear una carpeta una carpeta en C:/  
   Creo la carpeta **data** y dentro la carpeta **db**.
2. Ejecutar manualmente MongoDB:  
     
   Abro cmd.  
   Voy a la ruta donde está instalado MongoDB:  
   **cd C:\Program Files\MongoDB\Server\5.0\bin**  
   **mongod.exe**

Ahora en la carpeta db, tengo que tener ficheros que se han autogenerado.  
Si diera fallo, ejecutaría **mongod.exe --dbpath C:/data/db**

1. Para correr MongoDB, abro otro terminal y voy a la ruta  
   **cd C:\Program Files\MongoDB\Server\5.0\bin**  
   **mongo.exe**

**EJECUTAR MONGODB**

1. Abro cmd
2. **cd C:\Program Files\MongoDB\Server\5.0\bin**
3. **mongod.exe**
4. Abro otro cmd
5. **cd C:\Program Files\MongoDB\Server\5.0\bin**
6. **mongo.exe**

**COMANDOS MONGODB**

|  |  |
| --- | --- |
| Mostrar BBDD | **show dbs** |
| Mostrar colecciones de una BBDD | **show collections** |
| Seleccionar BBDD o collection | **use bbdd use collection** |
| Mostrar datos collection | **db.collection.find()** |
| Filtrar en una collection | **db.collection.find( { campo: "valor", … } )** |
| Borrar solo 1 registro de una collection que coincida | **db.collection.deleteOne( { campo: "valor", … } )** |
| Borrar TODOS los registros de una collection que coincidan | **db.collection.deleteMany( {campo: "valor", … } )** |
| Salir de la consola | **exit** |

Voy a crear la BBDD videojuegos  
**use videojuegos**

Dentro creo la colección PS4 y unos cuantos juegos.  
**db.ps4.save( { titulo: "The last of us", genero: "Aventuras", year: 2013 } )**

Miro lo que tiene la colección ps4  
**use videojuegos  
db.ps4.find()**

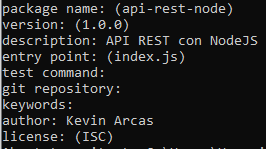
# **NodeJS**

## Instalaciones previas

Para hacer una API REST necesito tener descargados: NodeJS, MongoDB, Postman.

1. Descargar e instalar **NodeJS** LTS desde su web.  
   Comprobar que está todo instalado
   1. node --version 🡪 ver si node está instalado.
   2. npm --version 🡪 ver si npm está instalado.
2. Tener descargado e instalado **MongoDB**.
3. Descargar **Postman** desde su web para hacer pruebas.  
     
   Para hacer una prueba:  
     
   Le doy a **new** > **HTTP Request**  
   Pongo una **URL** de reqres.in y le doy a **send**.

## Crear proyecto NodeJS

1. Creo una carpeta en mi equipo.
2. Con la consola voy a esa ruta y ejecuto **npm init**.  
   
3. Abro el proyecto con VSCode

# **Express**

Una vez que tenemos creado el proyecto con NodeJS, vamos a usar Express, que es un Framework con el cual manejamos las peticiones HTTP (GET, POST, PUT).

## Instalaciones previas

Abro la terminal en VSCode, y voy a instalar las dependencias necesarias.

**npm i express --save** Con --save guarda la dependencia dentro del proyecto.

**npm i mongoose --save**  Instalo mongoose, que es una librería de MongoDB.

**npm i multer --save** Para subir archivos.

**npm i validator --save** Para validar datos.

**npm i cors --save** Para hacer peticiones entre dominios cruzados.

**npm i nodemon --save-dev** Watcher de cambios, para no parar y arrancar el server.

## Configuraciones previas

En el package.json, en la parte de scripts, voy a crear un “atajo” para ejecutar el archivo principal de la app en nodemon.



Creo el archivo index.js

Para correr la app con nodemon ejecuto en la consola **npm start**  
Si lo quiero correr sin pasar por nodemon, ejecuto **node index.js**