

# TRABAJANDO CON BASES DE DATOS: MIS PRIMEROS PASOS

María del Carmen Pareja Ferreira

En este trabajo, utilizaremos Power Shell y VS Code para la creación de una base de datos que recogerá información sobre dos proyectos musicales japoneses: Hypnosis Mic y Paradox Live. Dicha base de datos se compondrá de dos colecciones, una por cada proyecto musical, que contendrán información sobre cada uno de los integrantes y las bandas a las que pertenecen.



©Paradox Live2019

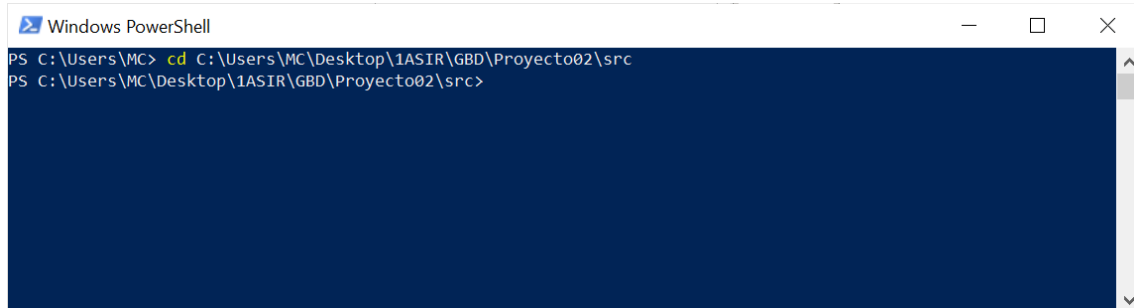
**-Gestión de base de datos-**

**1º ASIR**

# PASO 1: CREACIÓN DE UNA BASE DE DATOS

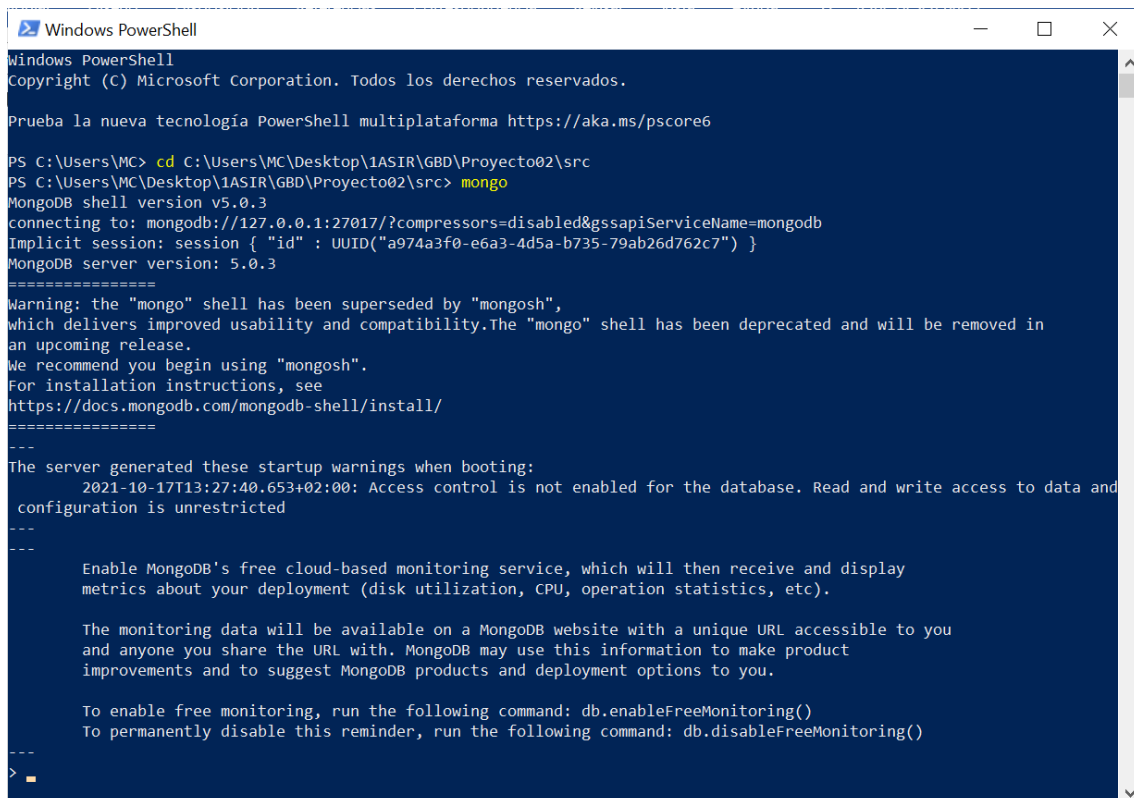
En este apartado, utilizaremos Power Shell para la creación de nuestra primera base de datos.

1. Usando el comando `> cd`, sitúese en el directorio donde se almacenará nuestra base de datos.



```
Windows PowerShell
PS C:\Users\MC> cd C:\Users\MC\Desktop\1ASIR\GBD\Proyecto02\src
PS C:\Users\MC\Desktop\1ASIR\GBD\Proyecto02\src>
```

2. Ejecute mongo mediante el comando `> mongo`.



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

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

PS C:\Users\MC> cd C:\Users\MC\Desktop\1ASIR\GBD\Proyecto02\src
PS C:\Users\MC\Desktop\1ASIR\GBD\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("a974a3f0-e6a3-4d5a-b735-79ab26d762c7") }
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Warning: the "mongo" shell has been superseded by "mongosh",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:
  2021-10-17T13:27:40.653+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()
---
> 
```

3. Para la creación de la base de datos, utilizaremos el comando `> use nombre`. En nuestro caso llamaremos a esta base de datos “jpmusic”.



```
Windows PowerShell
---
> use jpmusic
switched to db jpmusic
>
```

4. Escribiendo el comando `> db` podremos comprobar que efectivamente nos encontramos en dicha base de datos.

```
Windows PowerShell
switched to db jpmusic
> db
jpmusic
>
```

5. Ahora que sabemos que la base de datos se ha creado correctamente, crearemos la primera colección “hypmic” en su interior insertando directamente un documento formato JSON con información sobre uno de los integrantes. Para ello, utilizaremos el método `“insertOne()”`.

`> db.hypmic.insertOne({nombre: “Ichiro Yamada”, edad: 19, altura: 185, banda: {nombrebanda: “Buster Bros”, posicion: “lider”}})`

```
Windows PowerShell
> db.hypmic.insertOne({nombre: "Ichiro Yamada", edad: 19, altura: 185, banda: {nombrebanda: "Buster Bros", posicion: "lider"}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("616cb2eadd86733328c0f666")
}
>
```

La Shell nos devuelve un “true” para indicarnos que el documento se ha subido y la colección se ha actualizado.

6. Ahora crearemos la otra base de datos “paradoxlive” introduciendo otro documento JSON.

`> db.paradoxlive.insertOne({nombre: “Allen Sugasano”, edad: 21, altura: 185, banda: {nombrebanda: “BAE”, posicion: “lider”}})`

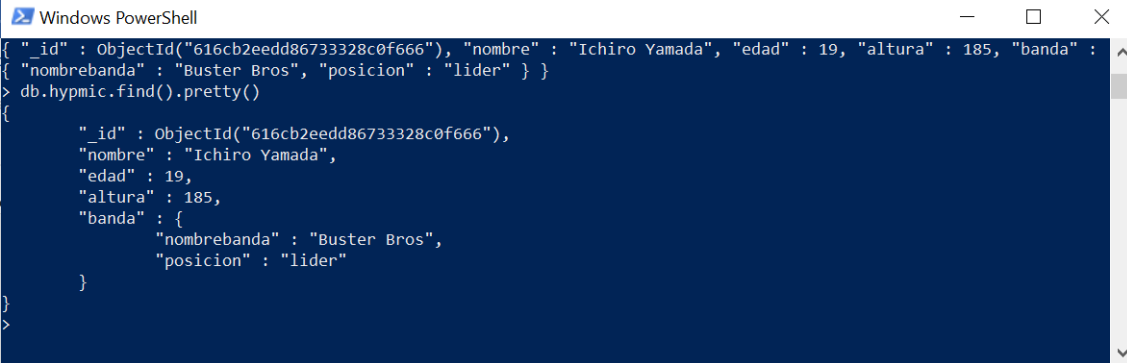
```
Windows PowerShell
> db.paradoxlive.insertOne({nombre: "Allen Sugasano", edad: 21, altura: 178, banda: {nombrebanda: "BAE", posicion: "lider"}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("616cb37cdd86733328c0f667")
}
>
```

7. Para comprobar que ambas colecciones se han creado, podemos usar el comando `> show collections`.

```
Windows PowerShell
> show collections
hypmic
paradoxlive
>
```

Efectivamente se han creado ambas colecciones.

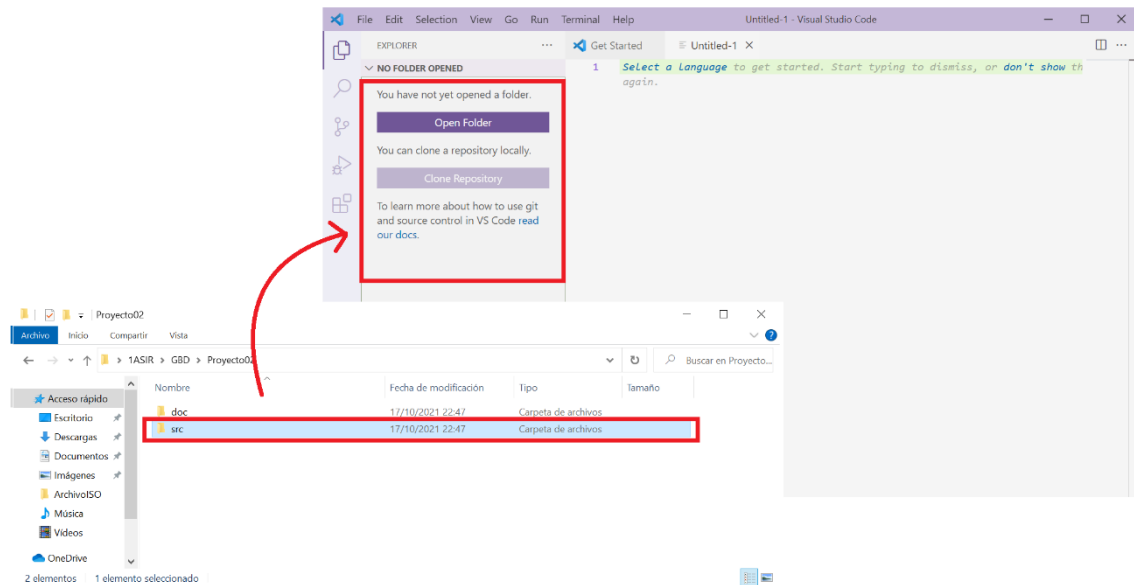
8. También podemos usar el método `find()` en una de nuestras colecciones para comprobar qué información se ha almacenado una colección. Si además añadimos el `pretty()` detrás, la información se dispondrá de manera más ordenada.



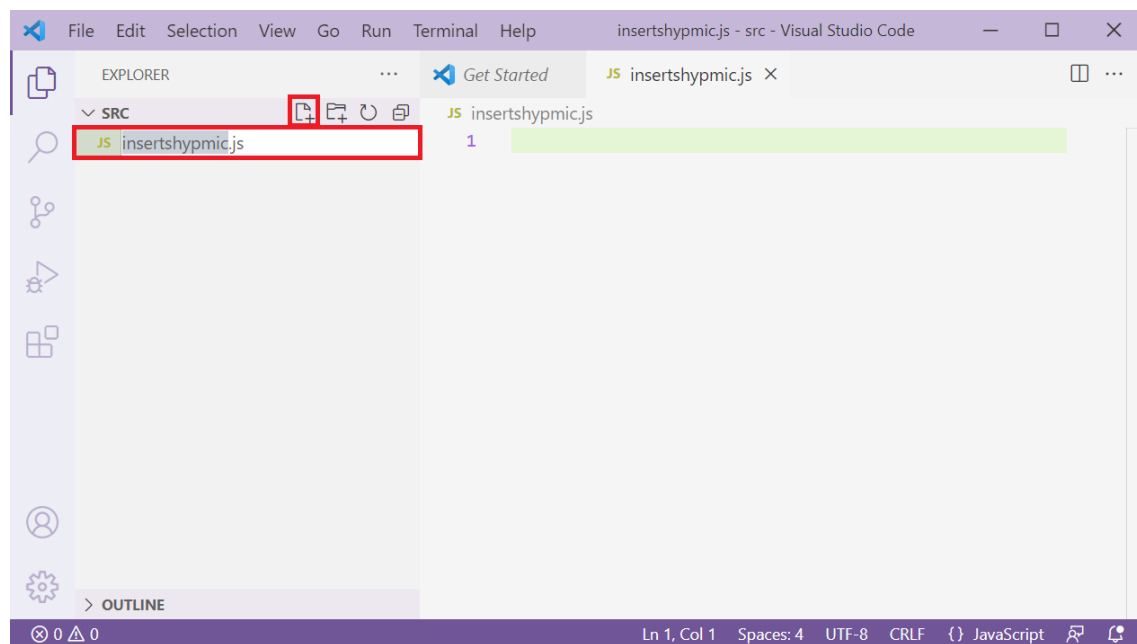
```
Windows PowerShell
{ "_id" : ObjectId("616cb2eadd86733328c0f666"), "nombre" : "Ichiro Yamada", "edad" : 19, "altura" : 185, "banda" :
{ "nombrebanda" : "Buster Bros", "posicion" : "lider" } }
> db.hypmic.find().pretty()
{
  "_id" : ObjectId("616cb2eadd86733328c0f666"),
  "nombre" : "Ichiro Yamada",
  "edad" : 19,
  "altura" : 185,
  "banda" : {
    "nombrebanda" : "Buster Bros",
    "posicion" : "lider"
  }
}
```

## PASO 2: CREACIÓN DE DOCUMENTOS JSON EN VISUAL STUDIO CODE

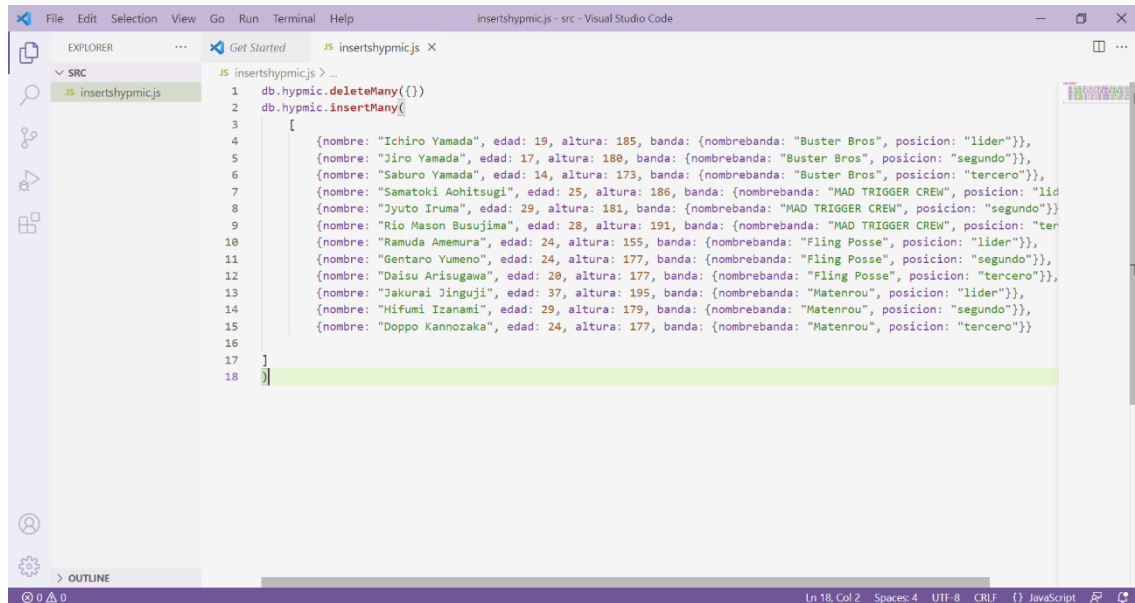
1. Pese a haber aprendido como introducir documentos JSON desde la shell, es mucho más dinámico hacerlo desde VS Code. En primer lugar, arrastraremos nuestra carpeta src a pantalla de VS Code.



2. Abrimos un archivo .js nuevo y le damos un nombre, en este caso "insertshypmic.js". Este archivo será posteriormente incluido en nuestra colección hypmic.

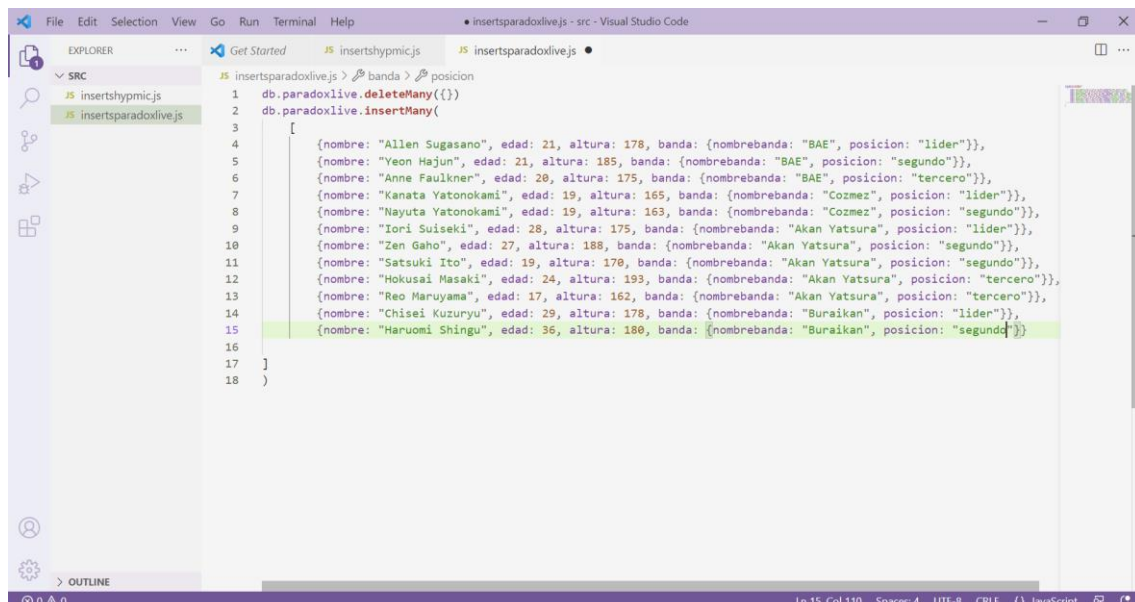


3. A continuación, añadiremos los documentos de todos los integrantes pertenecientes a la colección hypmic. En este caso se trata de más de un documento, por lo que en vez del método “insertOne()”, usaremos “.insertMany()”. Justo antes de iniciar la escritura, insertaremos el método “.deleteMany()” para eliminar todos los documentos que hayamos escrito anteriormente en la colección hypmic, de tal forma que ahora simplemente podamos agregarlos todos los inserts de una sentada.



```
1 db.hypmic.deleteMany({})
2 db.hypmic.insertMany([
3   {
4     nombre: "Ichiro Yamada", edad: 19, altura: 185, banda: {nombrebanda: "Buster Bros", posicion: "lider"}},
5     nombre: "Jiro Yamada", edad: 17, altura: 180, banda: {nombrebanda: "Buster Bros", posicion: "segundo"}},
6     nombre: "Saburo Yamada", edad: 14, altura: 173, banda: {nombrebanda: "Buster Bros", posicion: "tercero"}},
7     nombre: "Samatoki Aohitsugui", edad: 25, altura: 186, banda: {nombrebanda: "MAD TRIGGER CREW", posicion: "lider"}},
8     nombre: "Jyuto Iruma", edad: 29, altura: 181, banda: {nombrebanda: "MAD TRIGGER CREW", posicion: "segundo"}},
9     nombre: "Rio Mason Busujima", edad: 28, altura: 191, banda: {nombrebanda: "MAD TRIGGER CREW", posicion: "tercero"}},
10    nombre: "Ramuda Amemura", edad: 24, altura: 155, banda: {nombrebanda: "Fling Posse", posicion: "lider"}},
11    nombre: "Gentaro Yumeno", edad: 24, altura: 177, banda: {nombrebanda: "Fling Posse", posicion: "segundo"}},
12    nombre: "Daitsu Arisugawa", edad: 28, altura: 177, banda: {nombrebanda: "Fling Posse", posicion: "tercero"}},
13    nombre: "Jakurai Jinguji", edad: 37, altura: 195, banda: {nombrebanda: "Matenrou", posicion: "lider"}},
14    nombre: "Hifumi Izanami", edad: 29, altura: 179, banda: {nombrebanda: "Matenrou", posicion: "segundo"}},
15    nombre: "Doppo Kannozaaka", edad: 24, altura: 177, banda: {nombrebanda: "Matenrou", posicion: "tercero"}}
16  ]
17 )
18 ]
```

4. Ahora crearemos otro nuevo archivo llamado “insertsparadoxlive.js” e introduciremos los documentos siguiendo las recomendaciones anteriores.



```
1 db.paradoxlive.deleteMany({})
2 db.paradoxlive.insertMany([
3   {
4     nombre: "Allen Sugawano", edad: 21, altura: 178, banda: {nombrebanda: "BAE", posicion: "lider"}},
5     nombre: "Yeon Hajun", edad: 21, altura: 185, banda: {nombrebanda: "BAE", posicion: "segundo"}},
6     nombre: "Anne Faulkner", edad: 28, altura: 175, banda: {nombrebanda: "BAE", posicion: "tercero"}},
7     nombre: "Kanata Yatonokami", edad: 19, altura: 165, banda: {nombrebanda: "Cozmez", posicion: "lider"}},
8     nombre: "Nayuta Yatonokami", edad: 19, altura: 163, banda: {nombrebanda: "Cozmez", posicion: "segundo"}},
9     nombre: "Iori Suiseki", edad: 28, altura: 175, banda: {nombrebanda: "Akan Yatsura", posicion: "lider"}},
10    nombre: "Zen Gaho", edad: 27, altura: 188, banda: {nombrebanda: "Akan Yatsura", posicion: "segundo"}},
11    nombre: "Satsuki Ito", edad: 19, altura: 170, banda: {nombrebanda: "Akan Yatsura", posicion: "segundo"}},
12    nombre: "Hokusai Masaki", edad: 24, altura: 193, banda: {nombrebanda: "Akan Yatsura", posicion: "tercero"}},
13    nombre: "Reo Maruyama", edad: 17, altura: 162, banda: {nombrebanda: "Akan Yatsura", posicion: "tercero"}},
14    nombre: "Chisei Kuzuryu", edad: 29, altura: 178, banda: {nombrebanda: "Buraikan", posicion: "lider"}},
15    nombre: "Haruomi Shingu", edad: 36, altura: 180, banda: {nombrebanda: "Buraikan", posicion: "segundo"}}
16  ]
17 )
18 ]
```

5. Usamos “Ctrl + S” para guardar el archivo. A continuación, volvemos a la shell y a través del comando `> load` cargamos nuestros archivos, que se guardarán dentro de sus respectivas colecciones.

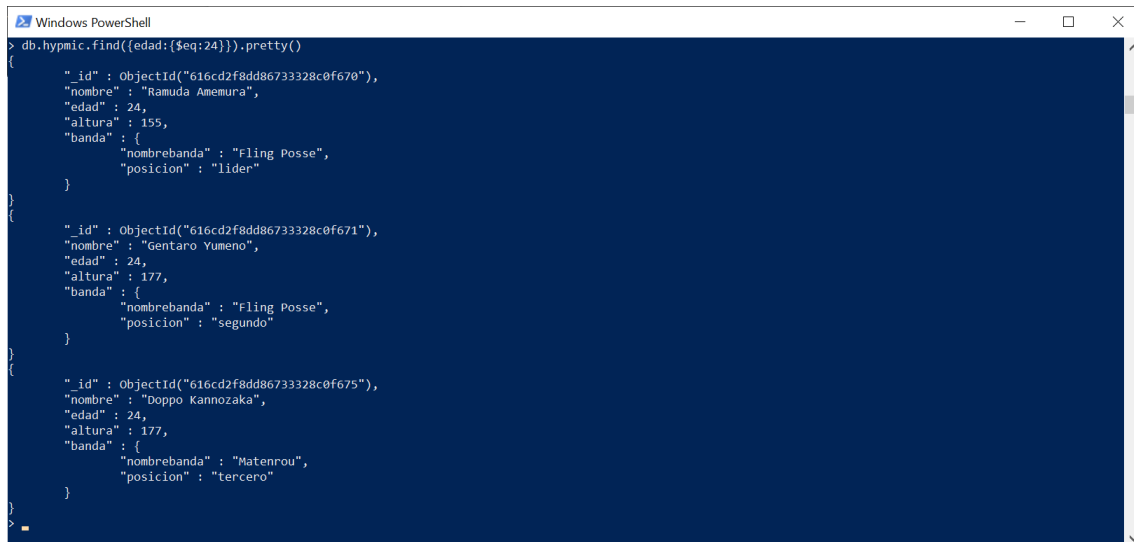
```
Windows PowerShell
> load("insertshypmic.js")
true
> load("insertsparadoxlive.js")
true
>
```

Podemos ejecutar un `“.find()”` para comprobar que nuestra base de datos se ha actualizado.

```
Windows PowerShell
> db.hypmic.find().pretty()
{
  "_id" : ObjectId("616cd2f8dd86733328c0f66a"),
  "nombre" : "Ichiro Yamada",
  "edad" : 19,
  "altura" : 185,
  "banda" : {
    "nombrebanda" : "Buster Bros",
    "posicion" : "lider"
  }
}
{
  "_id" : ObjectId("616cd2f8dd86733328c0f66b"),
  "nombre" : "Jiro Yamada",
  "edad" : 17,
  "altura" : 180,
  "banda" : {
    "nombrebanda" : "Buster Bros",
    "posicion" : "segundo"
  }
}
{
  "_id" : ObjectId("616cd2f8dd86733328c0f66c"),
  "nombre" : "Saburo Yamada",
  "edad" : 14,
  "altura" : 173,
  "banda" : {
    "nombrebanda" : "Buster Bros",
    "posicion" : "tercero"
  }
}
{
  "_id" : ObjectId("616cd2f8dd86733328c0f66d"),
  "nombre" : "Samatoki Aohitsugi",
  "edad" : 25,
  "altura" : 186,
  "banda" : {
    "nombrebanda" : "MAD TRIGGER CREW",
    "posicion" : "lider"
  }
}
```

## PASO 3: CONSULTAS EN LAS BASES DE DATOS

Añadiendo diversos selectores de consultas o query al método `“.find()”`, podemos ser capaces de localizar uno o varios archivos específicos a través de sus atributos. Uno de los query más usados es `“$eq”`, que especifica una condición de igualdad y busca documentos donde el valor de uno de sus campos sea igual valor especificado. Es tan común, que puede obviarse y no es necesario escribirlo cuando se especifica un valor.



```
Windows PowerShell
> db.hypmic.find({edad:$eq:24}).pretty()
{
  "_id" : ObjectId("616cd2f8dd86733328c0f670"),
  "nombre" : "Ramuda Amemura",
  "edad" : 24,
  "altura" : 155,
  "banda" : {
    "nombrebanda" : "Fling Posse",
    "posicion" : "lider"
  }
}
{
  "_id" : ObjectId("616cd2f8dd86733328c0f671"),
  "nombre" : "Gentaro Yumeno",
  "edad" : 24,
  "altura" : 177,
  "banda" : {
    "nombrebanda" : "Fling Posse",
    "posicion" : "segundo"
  }
}
{
  "_id" : ObjectId("616cd2f8dd86733328c0f675"),
  "nombre" : "Doppo Kannozaaka",
  "edad" : 24,
  "altura" : 177,
  "banda" : {
    "nombrebanda" : "Matenrou",
    "posicion" : "tercero"
  }
}
```

Al especificar que nos muestre aquellos integrantes cuya edad sea de 24 años, la shell únicamente nos mostrará los que cumplan tal requisito.

Entre los query más usados nos encontramos:

`$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.

Para finalizar, dejaré dos capturas más de consultas donde se utilicen algunos de los query expuestos anteriormente.



Consulta 1: encontrar a todos los miembros de la banda Akan Yatsura en la colección paradoxlife. Se usa el query \$eq.

```
Windows PowerShell
> db.paradoxlife.find({"banda.nombrebanda":{"$eq: "Akan Yatsura"}}).pretty()
{
  "_id" : ObjectId("616cd311dd86733328c0f67b"),
  "nombre" : "Iori Suiseki",
  "edad" : 28,
  "altura" : 175,
  "banda" : {
    "nombrebanda" : "Akan Yatsura",
    "posicion" : "lider"
  }
}
{
  "_id" : ObjectId("616cd311dd86733328c0f67c"),
  "nombre" : "Zen Gaho",
  "edad" : 27,
  "altura" : 188,
  "banda" : {
    "nombrebanda" : "Akan Yatsura",
    "posicion" : "segundo"
  }
}
{
  "_id" : ObjectId("616cd311dd86733328c0f67d"),
  "nombre" : "Satsuki Ito",
  "edad" : 19,
  "altura" : 170,
  "banda" : {
    "nombrebanda" : "Akan Yatsura",
    "posicion" : "segundo"
  }
}
{
  "_id" : ObjectId("616cd311dd86733328c0f67e"),
  "nombre" : "Hokusai Masaki",
  "edad" : 24,
  "altura" : 193,
  "banda" : {
    "nombrebanda" : "Akan Yatsura",
    "posicion" : "tercero"
  }
}
```

Consulta 2: encontrar a los miembros que midan más de 180cm en la colección hypmic. Se usa el query \$gt.

```
Windows PowerShell
> db.hypmic.find({"altura":{"$gt:180}}).pretty()
{
  "_id" : ObjectId("616cd2f8dd86733328c0f66a"),
  "nombre" : "Ichiro Yamada",
  "edad" : 19,
  "altura" : 185,
  "banda" : {
    "nombrebanda" : "Buster Bros",
    "posicion" : "lider"
  }
}
{
  "_id" : ObjectId("616cd2f8dd86733328c0f66d"),
  "nombre" : "Samatoki Aohitsugi",
  "edad" : 25,
  "altura" : 186,
  "banda" : {
    "nombrebanda" : "MAD TRIGGER CREW",
    "posicion" : "lider"
  }
}
{
  "_id" : ObjectId("616cd2f8dd86733328c0f66e"),
  "nombre" : "Jyuto Iruma",
  "edad" : 29,
  "altura" : 181,
  "banda" : {
    "nombrebanda" : "MAD TRIGGER CREW",
    "posicion" : "segundo"
  }
}
{
  "_id" : ObjectId("616cd2f8dd86733328c0f66f"),
  "nombre" : "Rio Mason Busujima",
  "edad" : 28,
  "altura" : 191,
  "banda" : {
    "nombrebanda" : "MAD TRIGGER CREW",
    "posicion" : "tercero"
  }
}
```