

MANUAL DE USUARIO “PRÁCTICA MICROSERVICIOS Y CONTENEDORES”

API REST DE CHISTES PARA CONTANAR EN UNA REUNIÓN



Ajitzi Ricardo Quintana Ruiz
8 de septiembre de 2020

Lenguaje de programación

El lenguaje de programación que elegí para hacer esta práctica es JavaScript, me parece un lenguaje muy versátil y amplia, con mucha documentación que te permite hacer cualquier cosa, es uno de mis favoritos y tengo la verificación de evaluaciones de LinkedIn en mi [perfil](#). Para hacerlo más eficaz utilizaré un framework para Node, Express, que facilita mucho el código para las API's y para el uso del protocolo HTTP.

Como base de datos usaré Mongo DB porque es una base de datos NoSQL que es basada en documentos y para aplicaciones sencillas orientadas a objetos es más que suficiente.



Estructura de mi API REST

Entrada

```
{
  "clave" : "maestro",
  "tipo" : "blanco",
  "longitud": "corto",
}
```

La entrada va a ser en formato JSON donde: clave, son las palabras que contenga el chiste; tipo, es tipo string con 2 opciones blanco o negro dependiendo de las personas en la reunión es el tipo de humor; longitud, son 2 opciones cortos (máximo 255 caracteres) y largos (más de 255 caracteres).

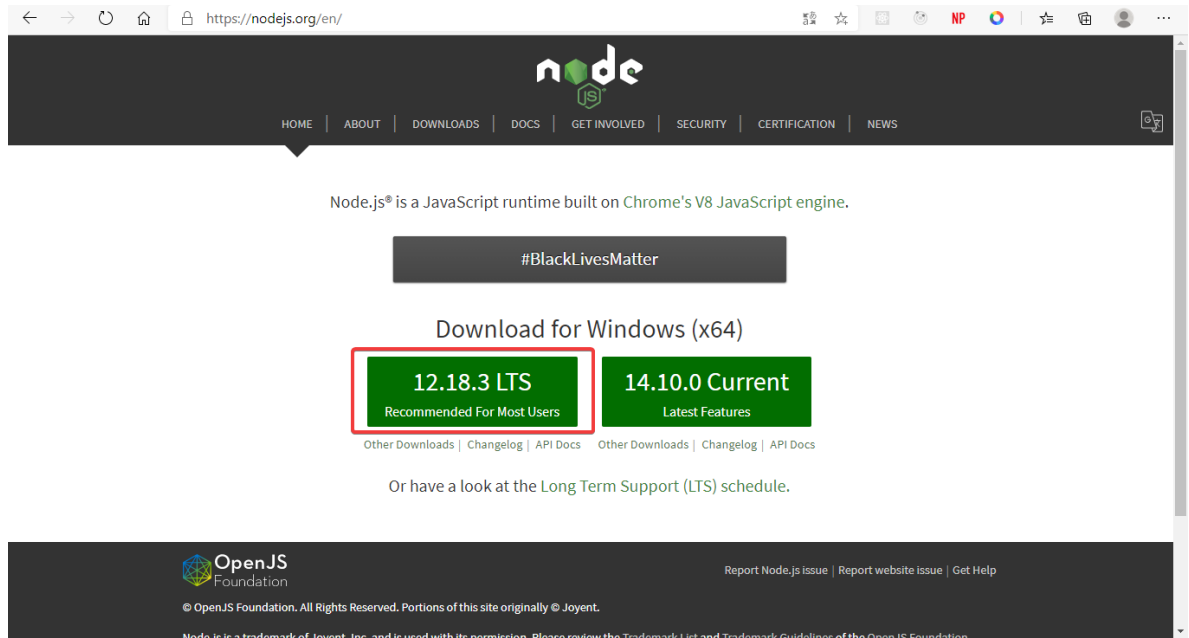
Salida

```
{
  "encontrado": true,
  "chistes": [
    {
      "chiste" : "- Andresito, ¿qué planeta va después de Marte? - Miércoles, maestro.",
      "tipo" : "blanco",
      "longitud": "corto",
      "gracia" : 5
    },
    {
      "chiste" : "- Mama, mama!!! En clase soy el mas alto y el que mas sabe!! - Claro cariño... eres el maestro...",
      "tipo" : "blanco",
      "longitud": "corto",
      "gracia" : 8
    }
  ]
}
```

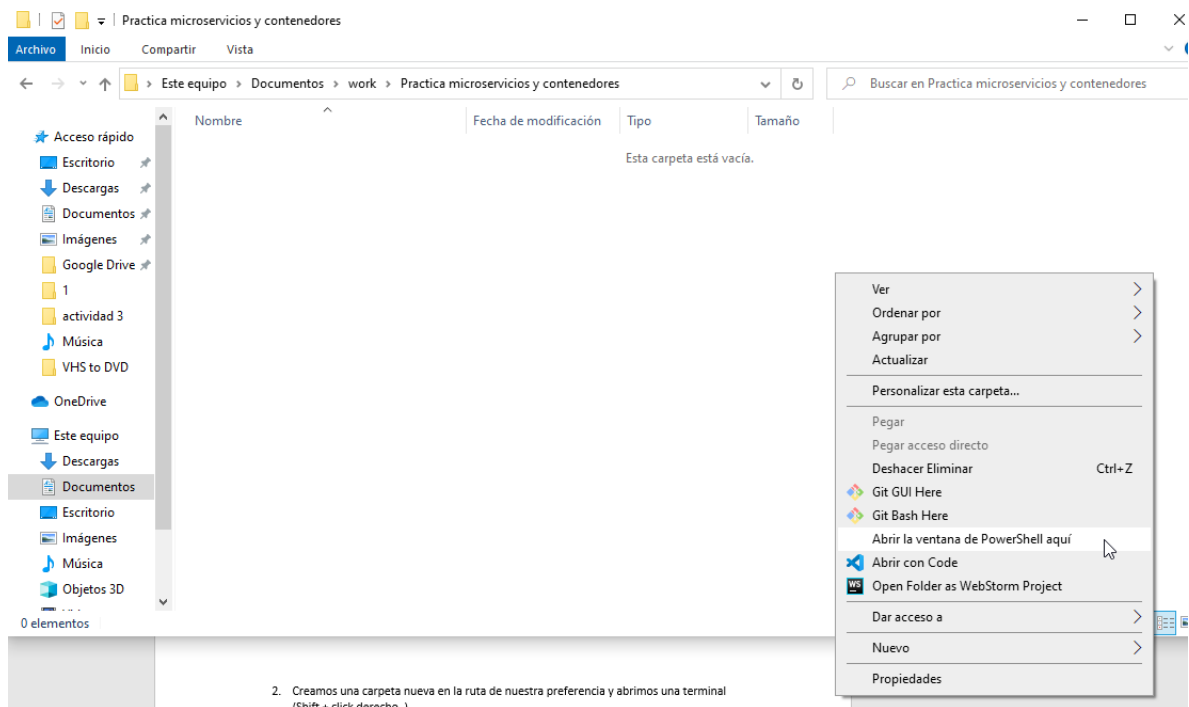
La salida va a ser en formato JSON con una propiedad, encontrado, que es un booleano que avisa si se encontraron chistes dependiendo de la entrada. Un arreglo de objetos donde vendrán los chistes encontrados donde: chiste, es un string con el chiste; tipo y longitud son igual que en la entrada y gracia que es un entero de 0 a 10 dependiendo de cuanta gracia le causo al que escribió el chiste en la base de datos, en este caso yo.

Desarrollo de la práctica

1. Descargamos e instalamos Node de la página [oficial](#). Es recomendable la versión LTS porque es más estable.



2. Creamos una carpeta en nuestra ruta de preferencia (el nombre de la carpeta debe ser sin espacios, estos se pueden sustituir con guiones). Abrimos una terminal (Shift + click derecho, abrir la ventana de PowerShell aquí).



3. Escribimos en la terminal el siguiente comando. Este lo que hará será crearnos el archivo package.json que contiene información de nuestra aplicación, nombre, descripción, dependencias, scripts, etc.

```
Windows PowerShell
PS C:\Users\ajitz\Documents\work\Practica_microservicios> npm init -y
Wrote to C:\Users\ajitz\Documents\work\Practica_microservicios\package.json:

{
  "name": "Practica_microservicios",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

PS C:\Users\ajitz\Documents\work\Practica_microservicios>
```

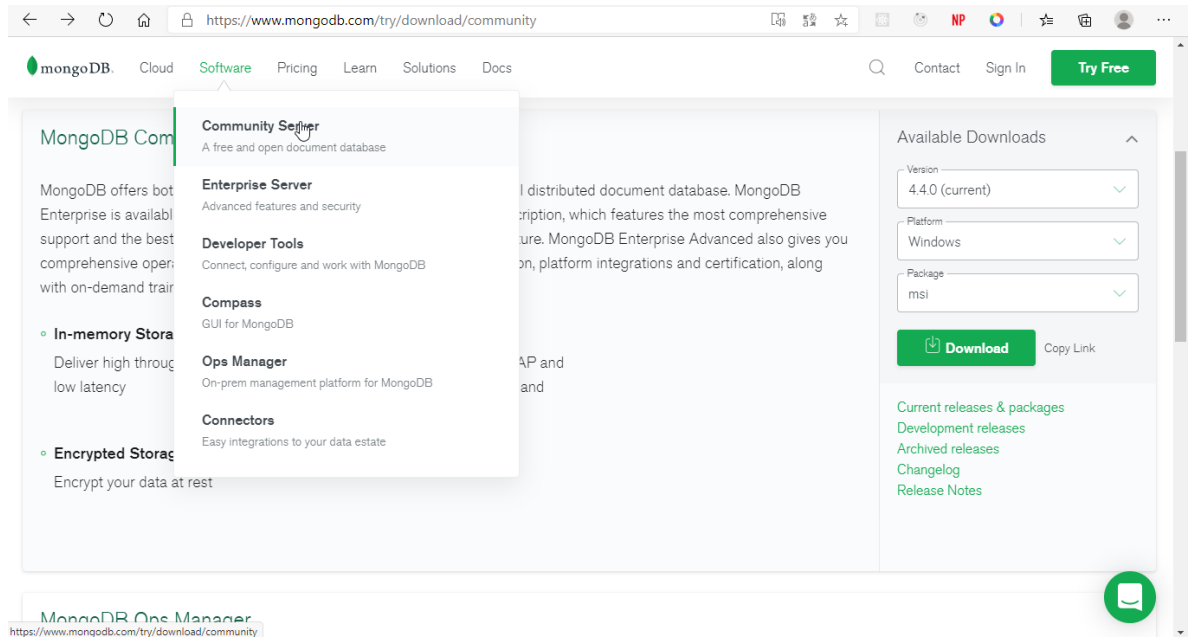
4. Instalamos nuestro framework Express con el siguiente comando, este comando se obtuvo de la pagina principal de [Express](#). Esto creara un archivo llamado package-lock.json qque contiene el orden en que se crearon las dependencias (**este archivo no debe ser editado manualmente**). Y una carpeta node_modules que contiene las dependencias para que nuestra aplicación se ejecute correctamente.

```
PS C:\Users\ajitz\Documents\work\Practica_microservicios> npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN Practica_microservicios@1.0.0 No description
npm WARN Practica_microservicios@1.0.0 No repository field.

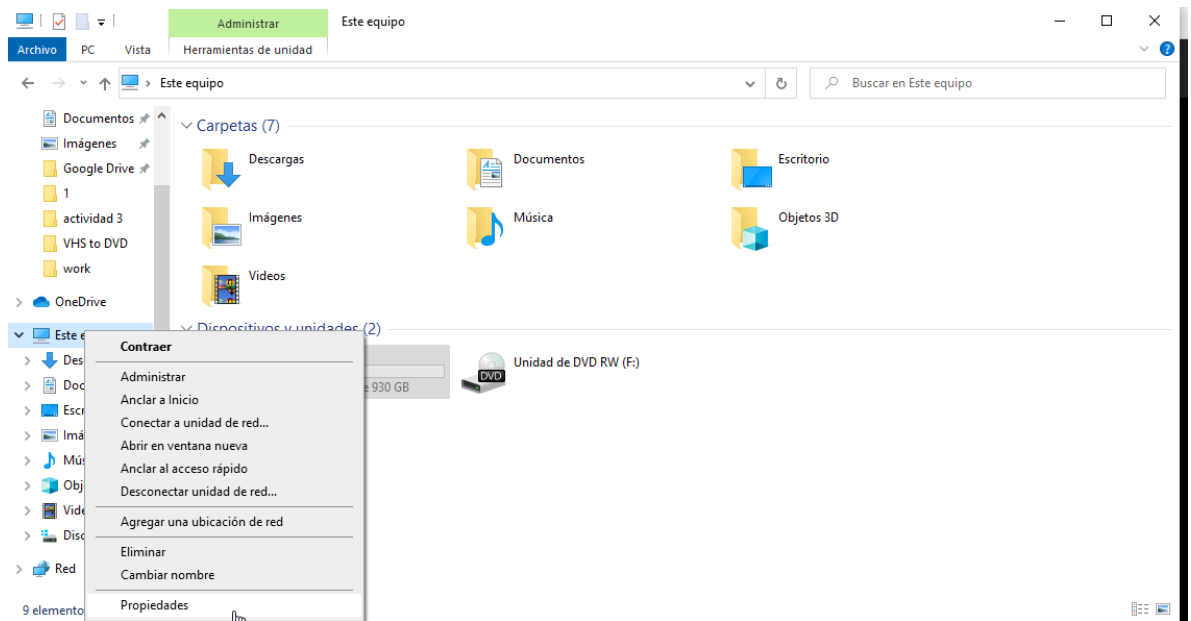
+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 10.14s
found 0 vulnerabilities

PS C:\Users\ajitz\Documents\work\Practica_microservicios>
```

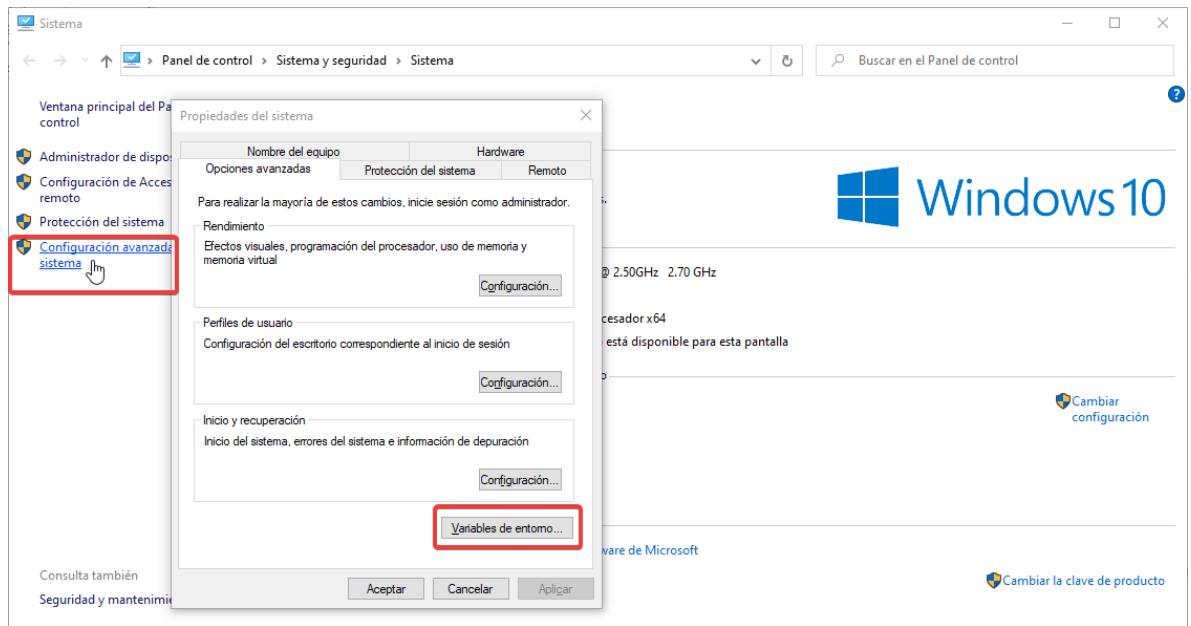
5. Descargamos Mongo DB Server de su [página oficial](#), instalamos la versión completa y Mongo Compass que es una interfaz para nuestra base de datos.



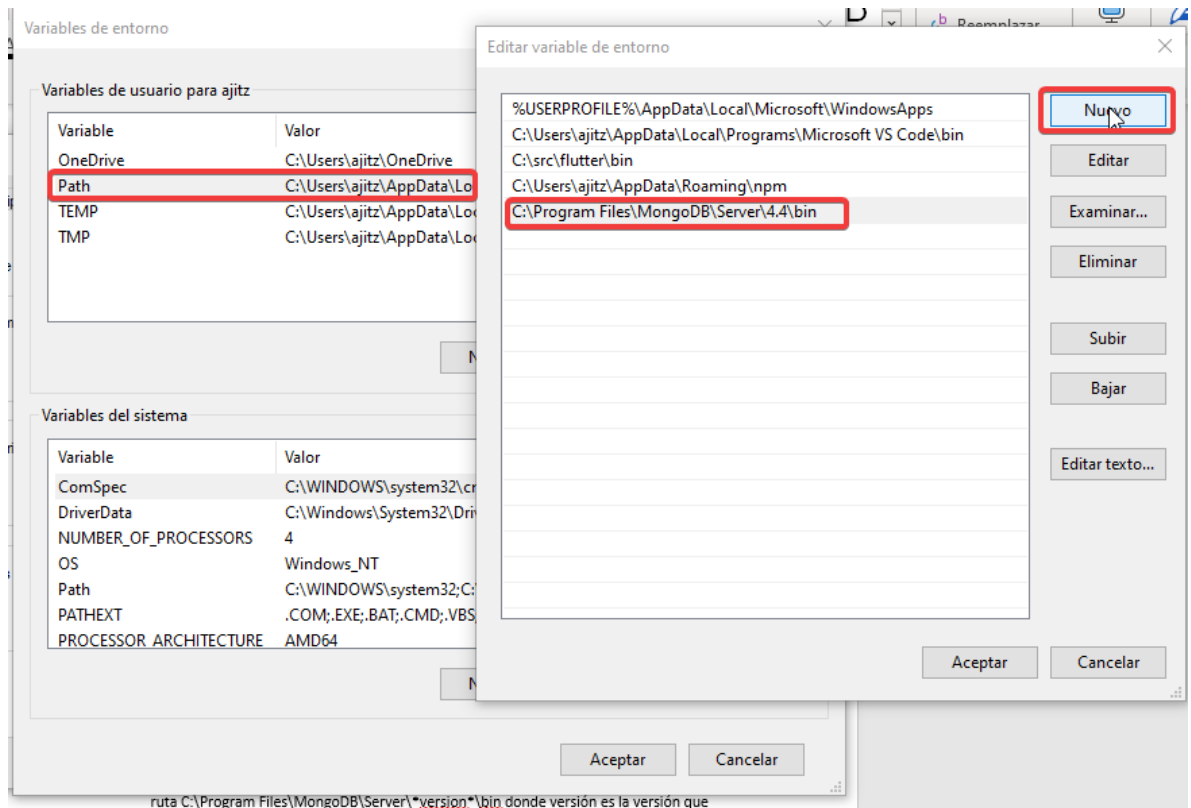
6. Ahora agregaremos la variable de entorno a nuestra computadora. Navegamos hasta la ruta C:\Program Files\MongoDB\Server*version*\bin donde versión es la versión que descargaron. Y copiamos dicha ruta, abrimos el explorador de archivos hacemos click derecho en este equipo y seleccionamos propiedades



Despues nos vamos a configuración avanzada del sistema -> variables de entorno

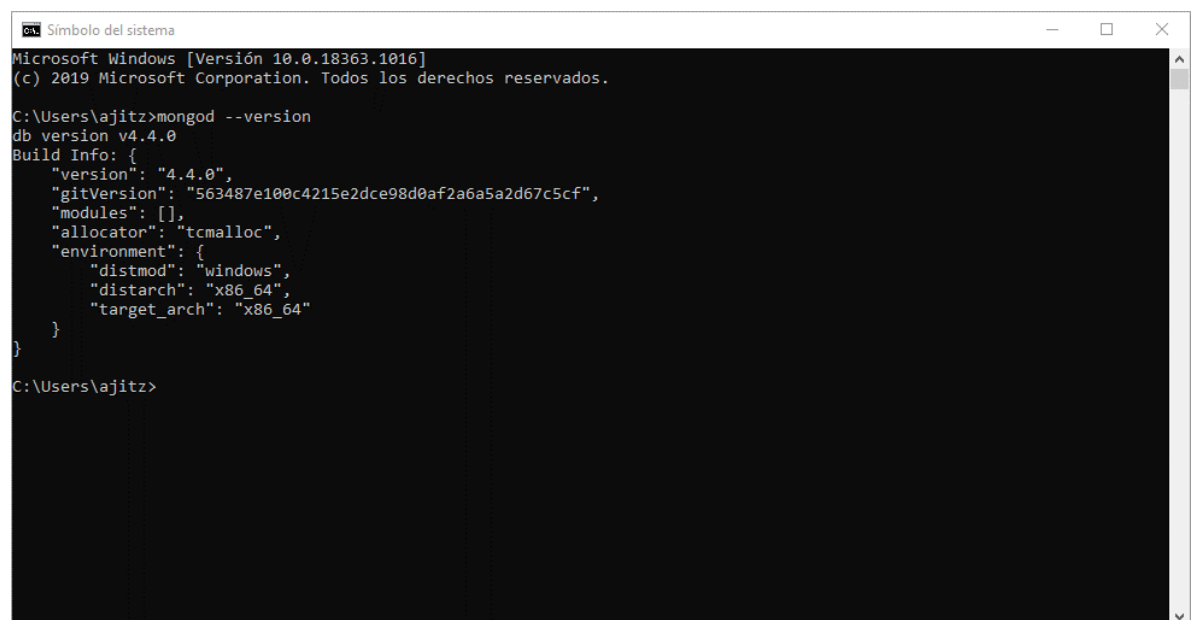
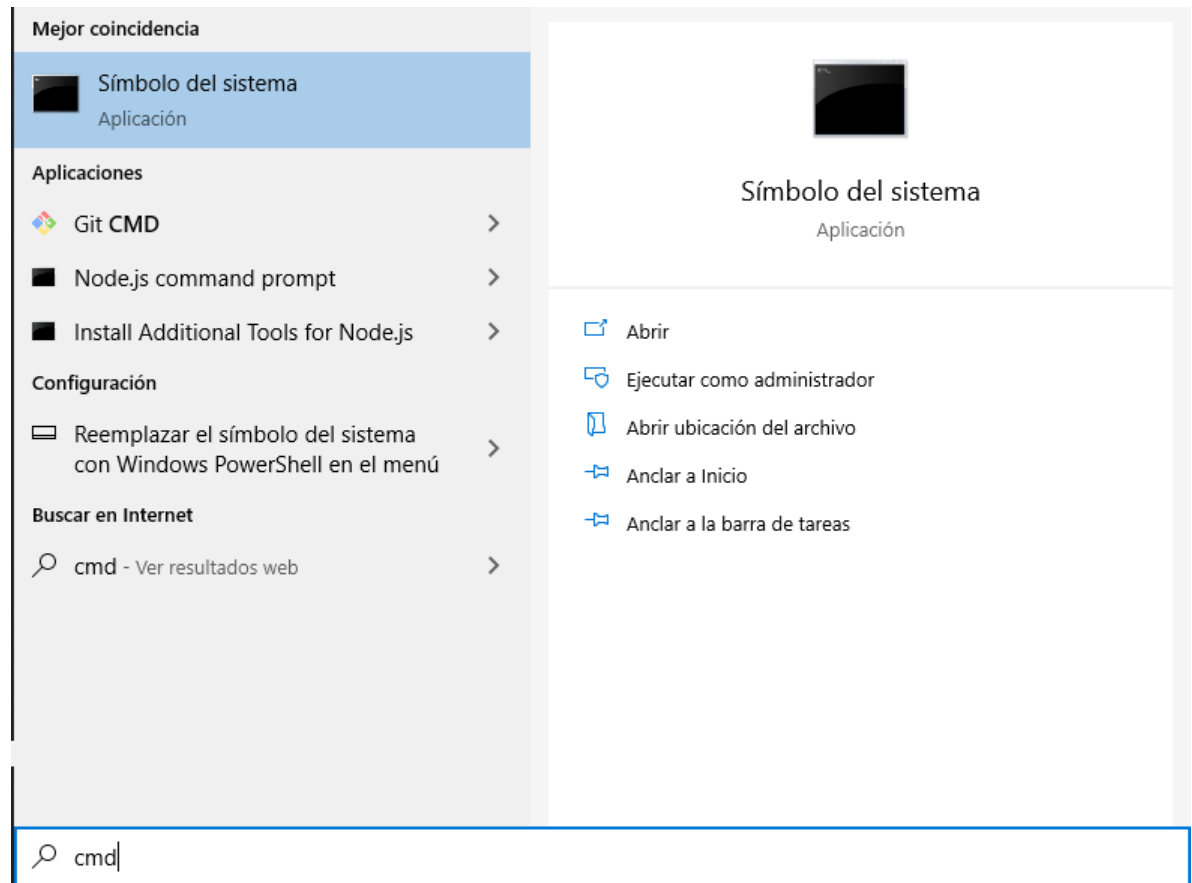


Hacemos click en path -> nuevo, pegamos la ruta y le damos en aceptar a todas las ventanas anteriores. Esto nos permitira levantar el servidor de Mongo DB en la terminal sin importar en la carpeta que estemos.

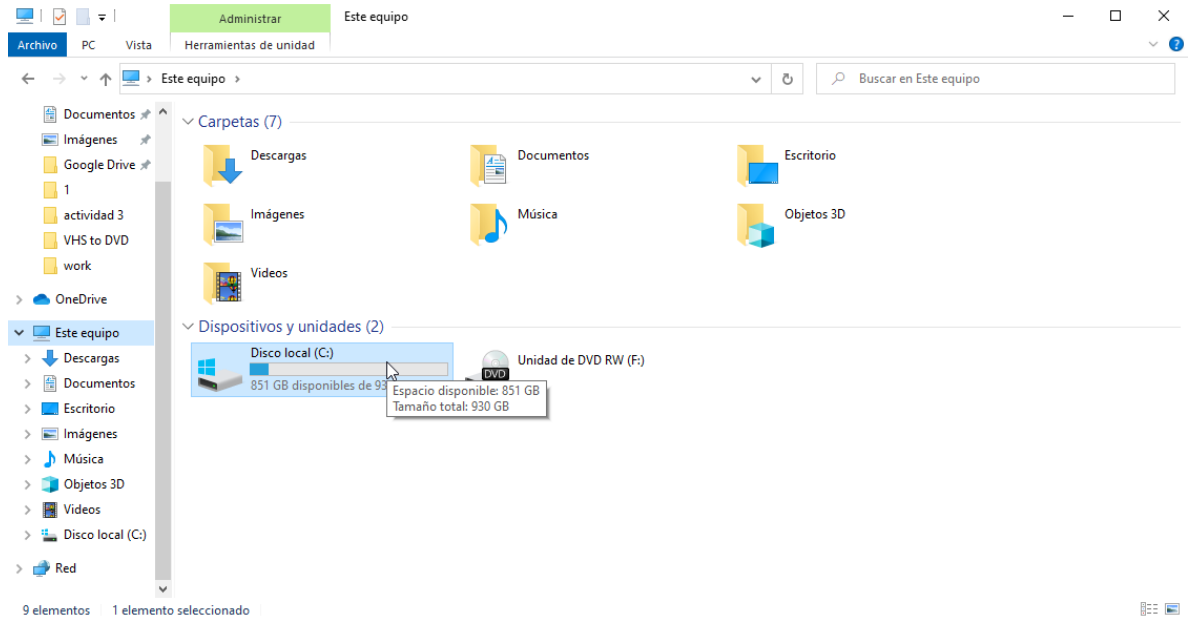


Para comprobarlo abrimos una terminal buscando cmd en el inicio y corremos el siguiente

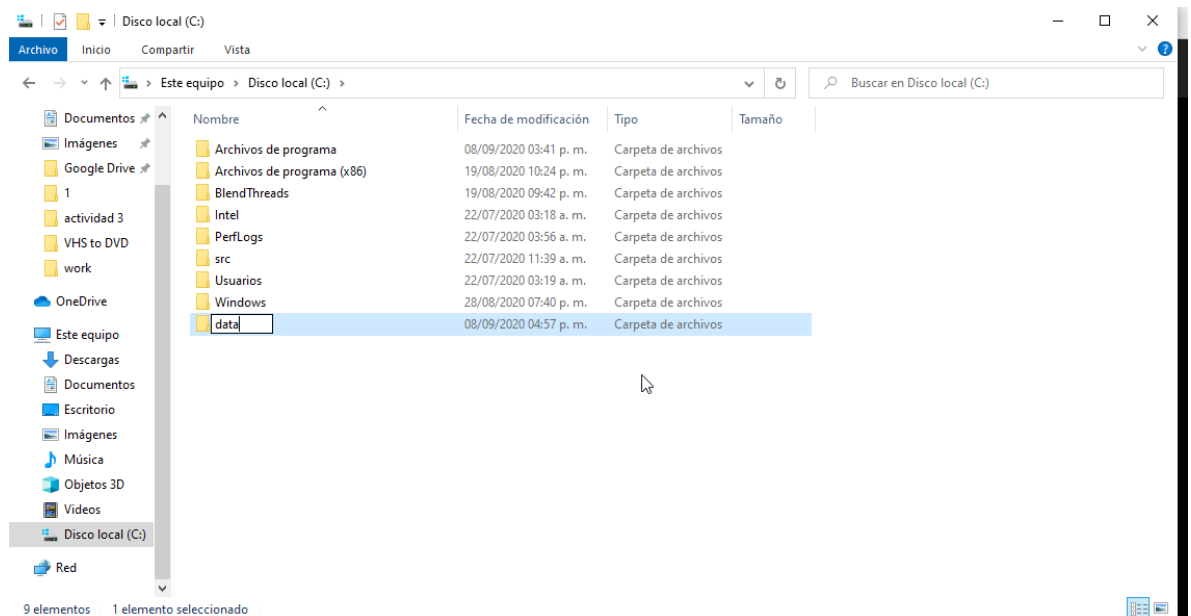
comando, si nos aparece esto quiere decir que creamos la variable de entorno correctamente.

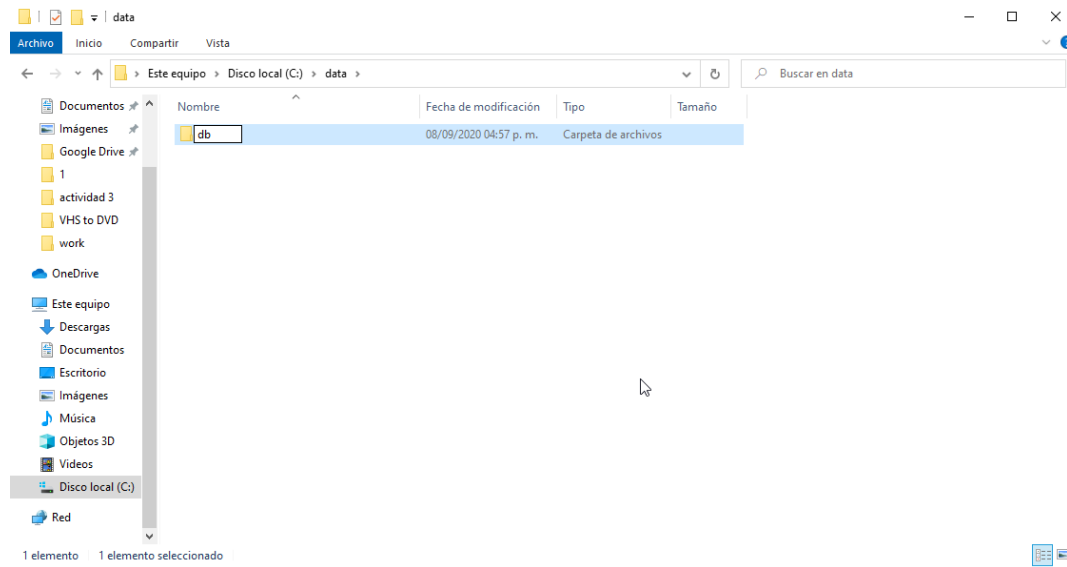


7. Abrimos un explorador de archivos -> este equipo -> disco local C:

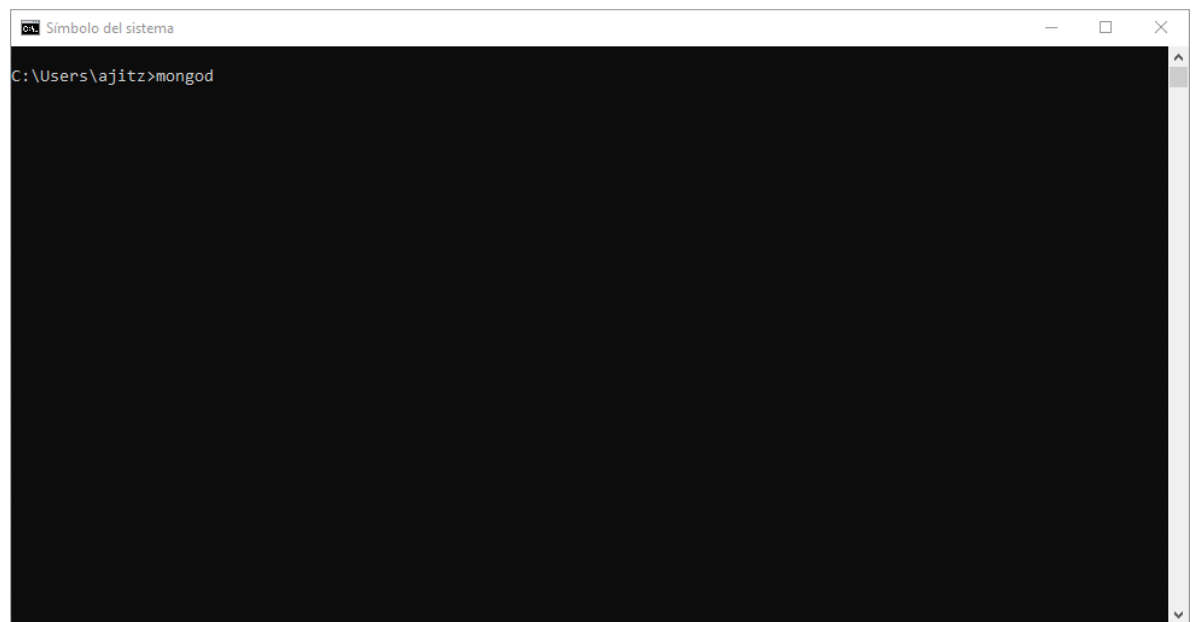


creamos una carpeta con nombre data, abrimos esta carpeta y creamos una nueva carpeta con nombre db





8. Abrimos otra terminal cmd como anteriormente lo realizamos y ejecutamos el siguiente comando, esto levantara el servicio de Mongo DB Server



9. Abrimos otra ventana cmd y ejecutamos el siguiente comando, damos enter y nos aparecerá lo siguiente

```
Símbolo del sistema - mongo
C:\Users\ajitz>mongo
MongoDB shell version v4.4.0
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("5896ddac-2921-4cd6-98f7-d7611b7bfa09") }
MongoDB server version: 4.4.0
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
  https://community.mongodb.com
---
The server generated these startup warnings when booting:
  2020-09-08T15:45:48.755-05:00: ***** SERVER RESTARTED *****
  2020-09-08T15:45:53.374-05:00: 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).

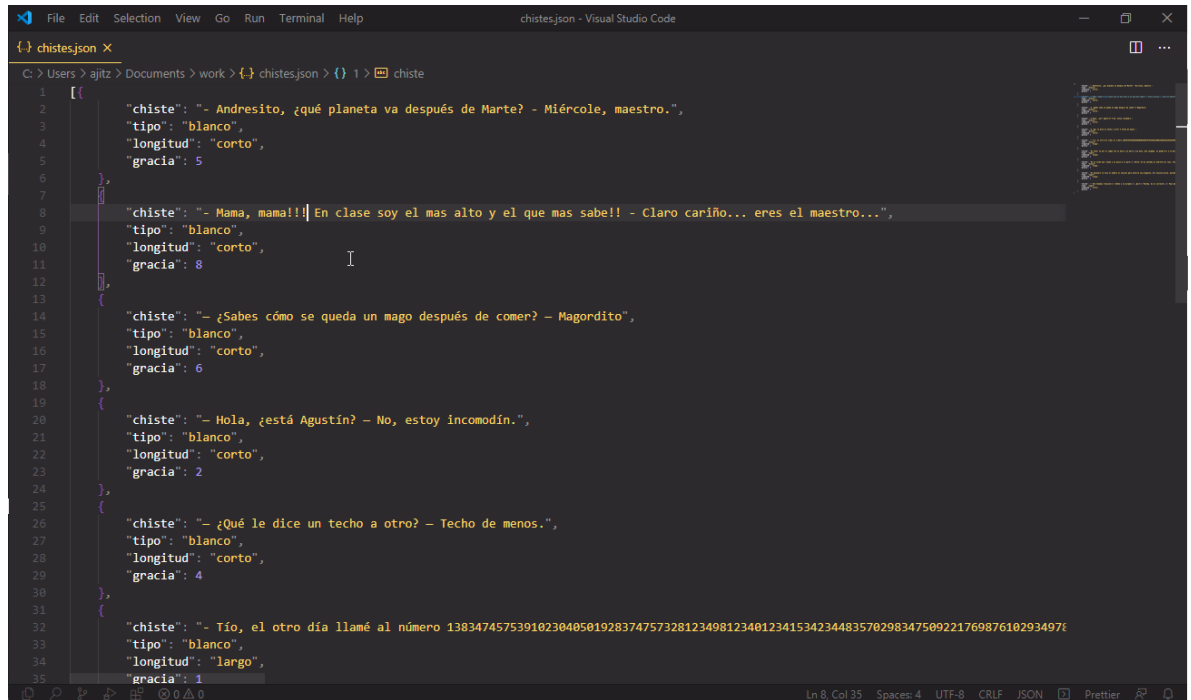
  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
  and anyone you share the URL with. MongoDB may use this information to make product
  improvements and to suggest MongoDB products and deployment options to you.

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

10. Ejecutamos los siguientes comandos en esta consola, use es para crear nuestra base de datos y createCollection para crear una nueva colección donde estarán nuestros chistes

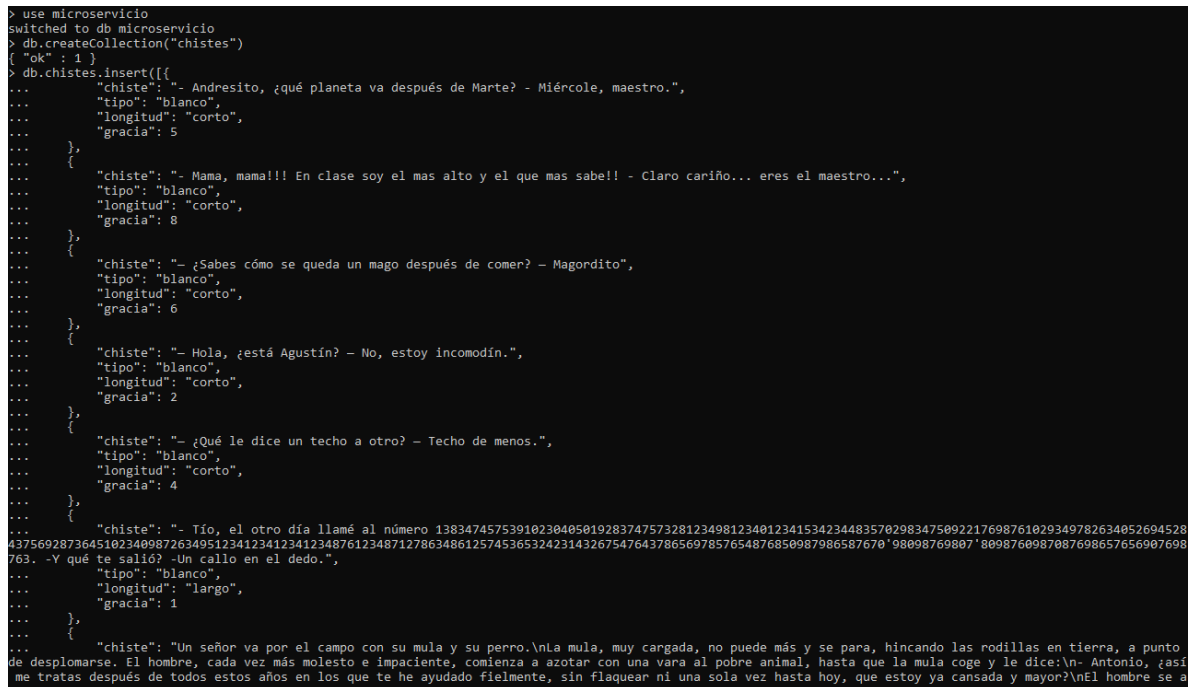
```
Símbolo del sistema - mongo
> use microservicio
switched to db microservicio
> db.createCollection("chistes")
{ "ok" : 1 }
>
```

11. Creamos un JSON con un arreglo con los chistes que queremos insertar en nuestra base de datos, yo voy a usar [Visual Studio Code](#) para hacerlo mas sencillo y más visible. escribiré pocos chistes porque es una prueba, todos serán sacados de internet. (El archivo chistes.json estará en el repositorio también).



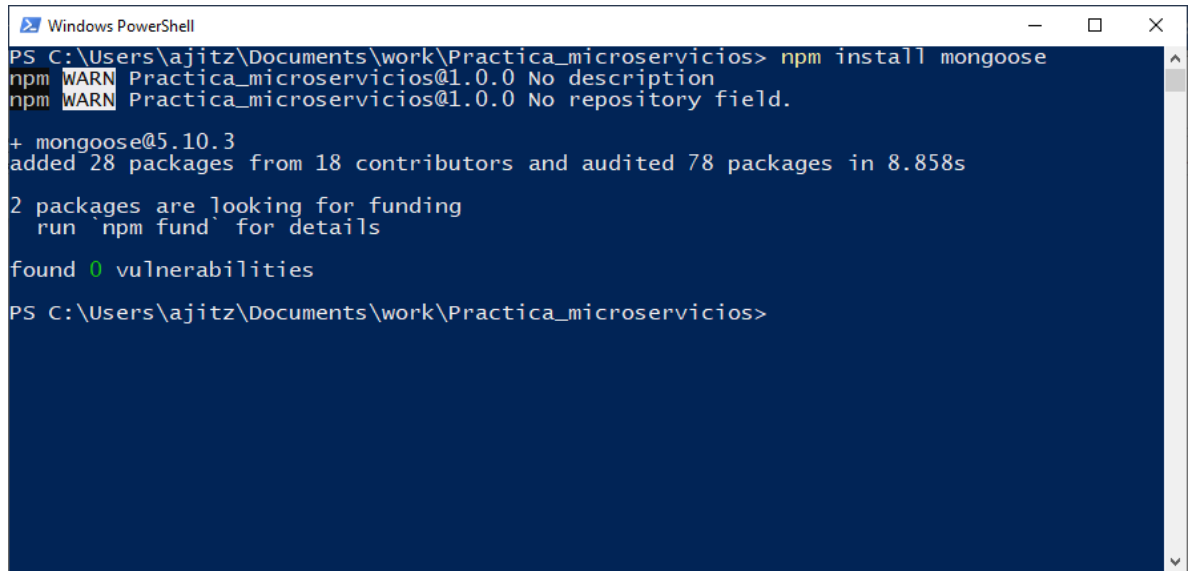
```
1 [{
2   "chiste": "- Andresito, ¿qué planeta va después de Marte? - Miércoles, maestro.",
3   "tipo": "blanco",
4   "longitud": "corto",
5   "gracia": 5
6 },
7
8   "chiste": "- Mama, mama!!! En clase soy el mas alto y el que mas sabe!! - Claro cariño... eres el maestro...",
9   "tipo": "blanco",
10  "longitud": "corto",
11  "gracia": 8
12 },
13
14  "chiste": "- ¿Sabes cómo se queda un mago después de comer? - Magordito",
15  "tipo": "blanco",
16  "longitud": "corto",
17  "gracia": 6
18 },
19
20  "chiste": "- Hola, ¿está Agustín? - No, estoy incomodín.",
21  "tipo": "blanco",
22  "longitud": "corto",
23  "gracia": 2
24 },
25
26  "chiste": "- ¿Qué le dice un techo a otro? - Techo de menos.",
27  "tipo": "blanco",
28  "longitud": "corto",
29  "gracia": 4
30 },
31
32  "chiste": "- Tío, el otro día llamé al número 13834745753910230405019283747573281234981234012341534234483570298347509221769876102934976102340987263495123412341234876123487127863486125745365324231432675476437865697857654876850987986587670'98098769807'8098760987087698657656907698763. -Y qué te salió? -Un callo en el dedo.",
33  "tipo": "blanco",
34  "longitud": "largo",
35  "gracia": 1
36 },
37
38  "chiste": "Un señor va por el campo con su mula y su perro.\nla mula, muy cargada, no puede más y se para, hincando las rodillas en tierra, a punto de desplomarse. El hombre, cada vez más molesto e impaciente, comienza a azotar con una vara al pobre animal, hasta que la mula coge y le dice:\n- Antonio, ¿así me tratas después de todos estos años en los que te he ayudado fielmente, sin flaquear ni una sola vez hasta hoy, que estoy ya cansada y mayor?\nEl hombre se a
```

12. En la consola que estábamos escribiendo `db.chistes.insert()` y entre los paréntesis pegamos el contenido del archivo de chistes.json



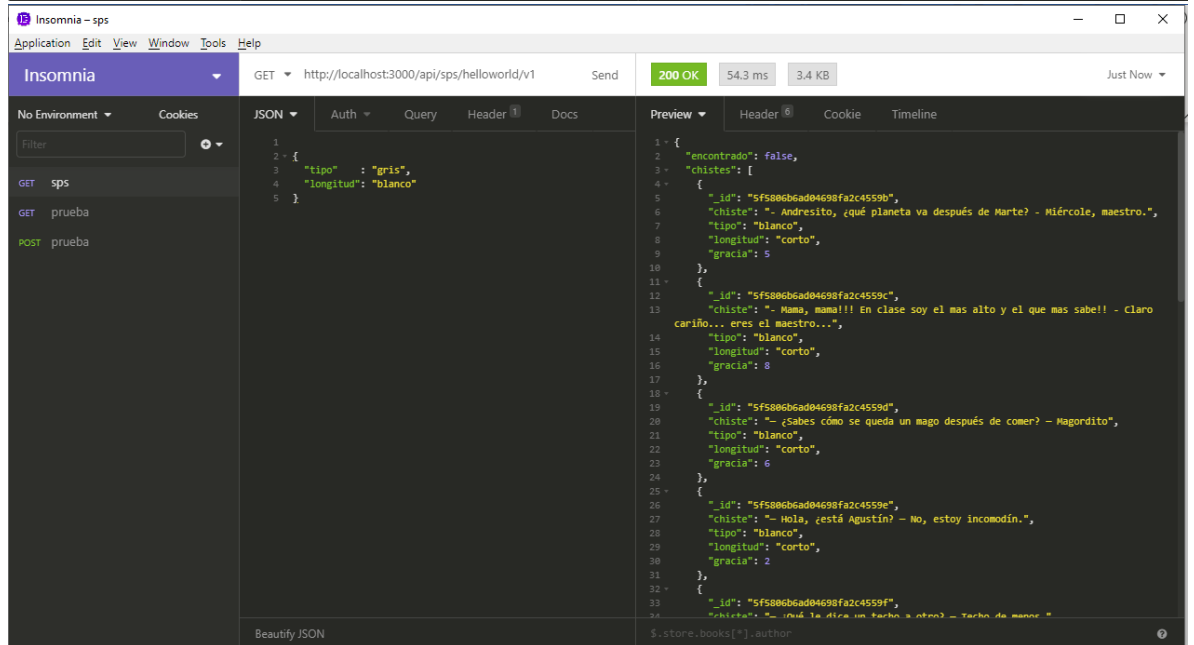
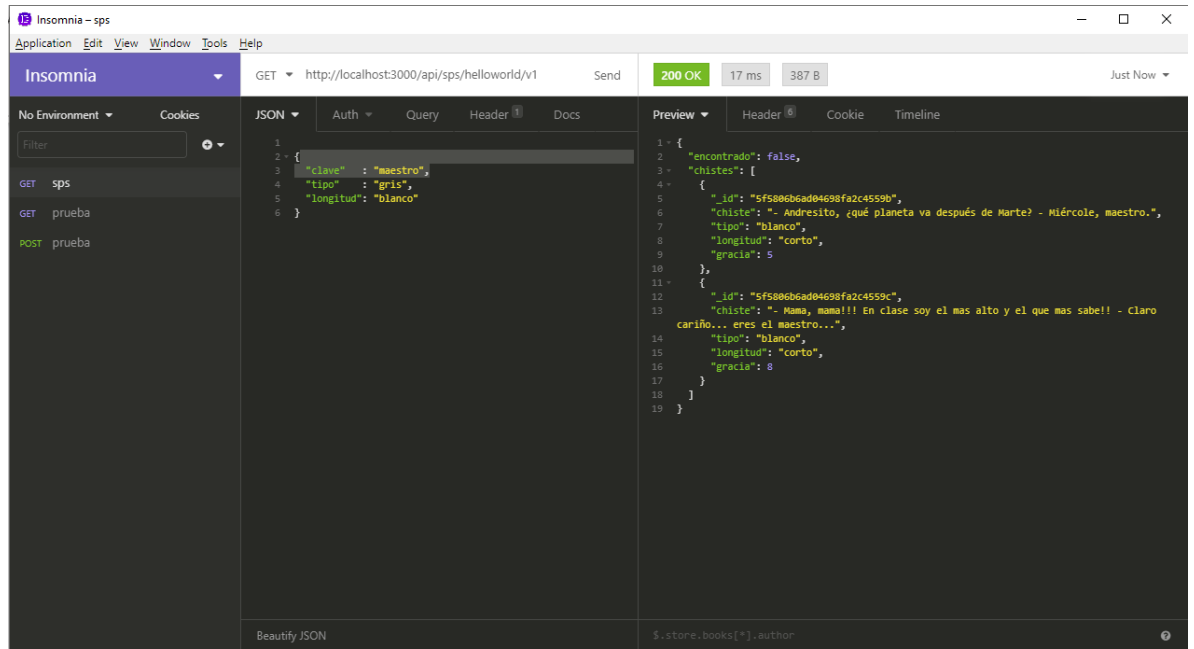
```
> use microservicio
switched to db microservicio
> db.createCollection("chistes")
{ "ok" : 1 }
> db.chistes.insert([
...   "chiste": "- Andresito, ¿qué planeta va después de Marte? - Miércoles, maestro.",
...   "tipo": "blanco",
...   "longitud": "corto",
...   "gracia": 5
... },
... {
...   "chiste": "- Mama, mama!!! En clase soy el mas alto y el que mas sabe!! - Claro cariño... eres el maestro...",
...   "tipo": "blanco",
...   "longitud": "corto",
...   "gracia": 8
... },
... {
...   "chiste": "- ¿Sabes cómo se queda un mago después de comer? - Magordito",
...   "tipo": "blanco",
...   "longitud": "corto",
...   "gracia": 6
... },
... {
...   "chiste": "- Hola, ¿está Agustín? - No, estoy incomodín.",
...   "tipo": "blanco",
...   "longitud": "corto",
...   "gracia": 2
... },
... {
...   "chiste": "- ¿Qué le dice un techo a otro? - Techo de menos.",
...   "tipo": "blanco",
...   "longitud": "corto",
...   "gracia": 4
... },
... {
...   "chiste": "- Tío, el otro día llamé al número 13834745753910230405019283747573281234981234012341534234483570298347509221769876102934976102340987263495123412341234876123487127863486125745365324231432675476437865697857654876850987986587670'98098769807'8098760987087698657656907698763. -Y qué te salió? -Un callo en el dedo.",
...   "tipo": "blanco",
...   "longitud": "largo",
...   "gracia": 1
... },
... {
...   "chiste": "Un señor va por el campo con su mula y su perro.\nla mula, muy cargada, no puede más y se para, hincando las rodillas en tierra, a punto de desplomarse. El hombre, cada vez más molesto e impaciente, comienza a azotar con una vara al pobre animal, hasta que la mula coge y le dice:\n- Antonio, ¿así me tratas después de todos estos años en los que te he ayudado fielmente, sin flaquear ni una sola vez hasta hoy, que estoy ya cansada y mayor?\nEl hombre se a
```

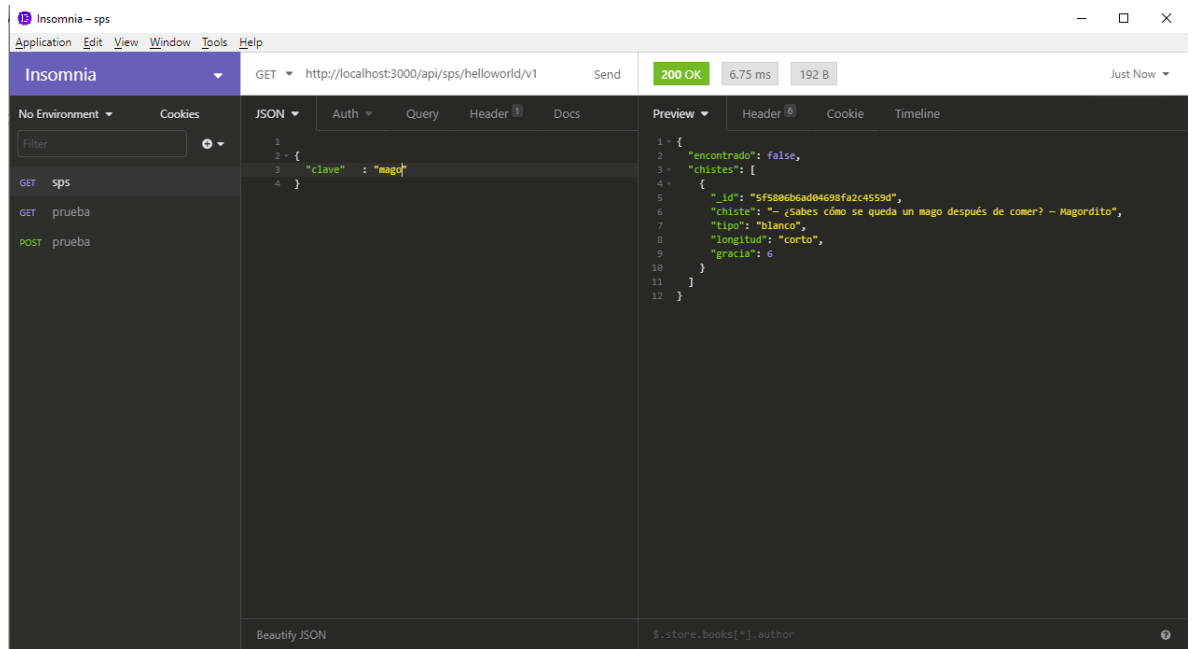
13. Abrimos la terminal dentro de nuestro proyecto y ejecutamos el comando `npm install mongoose` este es un paquete que necesitamos para comunicar nuestra base de datos con nuestra aplicación

A screenshot of a Windows PowerShell terminal window. The title bar at the top says "Windows PowerShell" and has standard window controls (minimize, maximize, close). The terminal background is dark blue. The text in the terminal is white. It shows the command `npm install mongoose` being executed. There are two warning messages from npm: `WARN Practica_microservicios@1.0.0 No description` and `WARN Practica_microservicios@1.0.0 No repository field.`. Then it shows the installation of `mongoose@5.10.3`, stating it added 28 packages from 18 contributors and audited 78 packages in 8.858s. It also mentions that 2 packages are looking for funding and to run `npm fund` for details. Finally, it says `found 0 vulnerabilities`. The prompt `PS C:\Users\ajitz\Documents\work\Practica_microservicios>` is visible at the bottom.

```
Windows PowerShell
PS C:\Users\ajitz\Documents\work\Practica_microservicios> npm install mongoose
npm WARN Practica_microservicios@1.0.0 No description
npm WARN Practica_microservicios@1.0.0 No repository field.
+ mongoose@5.10.3
added 28 packages from 18 contributors and audited 78 packages in 8.858s
2 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\ajitz\Documents\work\Practica_microservicios>
```

14. Creare el código para que se ejecute nuestra aplicación y la documentare dentro del mismo.
15. Instalamos cualquier programa que nos permita hacer peticiones http, a mi me gusta más insomnia que otros programas porque me parece mas minimalista pero cualquiera funcionaria. Hacemos una petición get a nuestro localhost con el url requerido, en el cuerpo de la petición ponemos JSON llenamos los datos y le damos enviar. Podemos ver que nuestra búsqueda si cambia dependiendo los valores de entrada.





16. Instalamos [docker](https://www.docker.com/get-started)

