

Enlaces:

<https://nodejs.org/es/>

Node.js® es un entorno de ejecución para JavaScript construido con [V8, motor de JavaScript de Chrome](#).



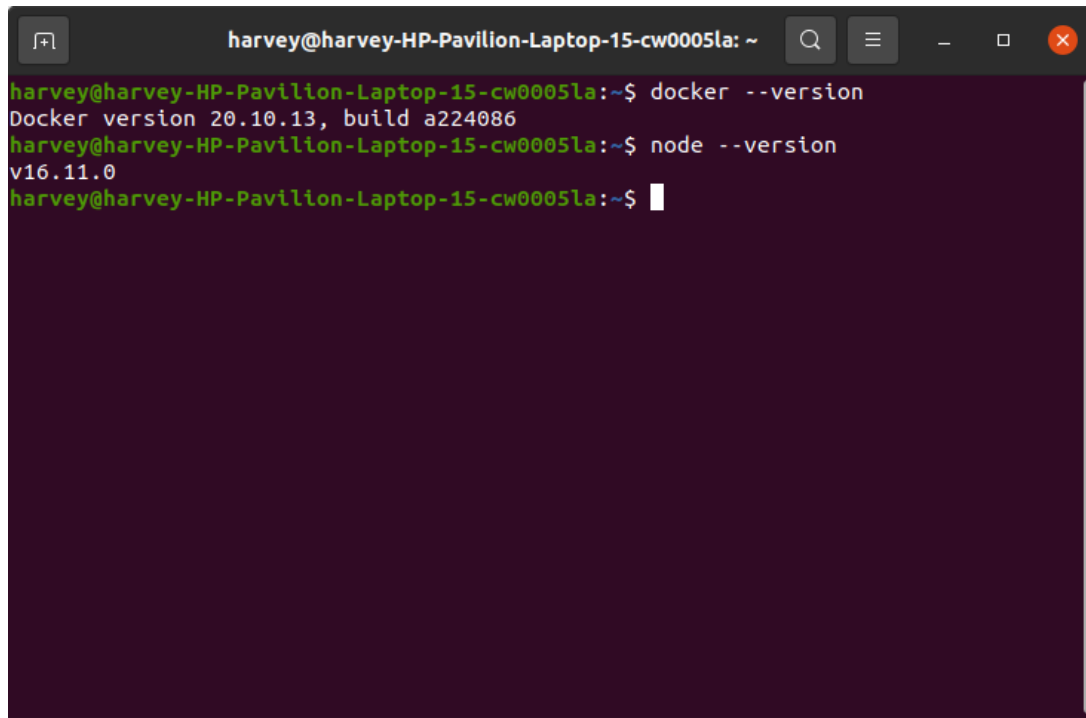
MariaDB

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL.



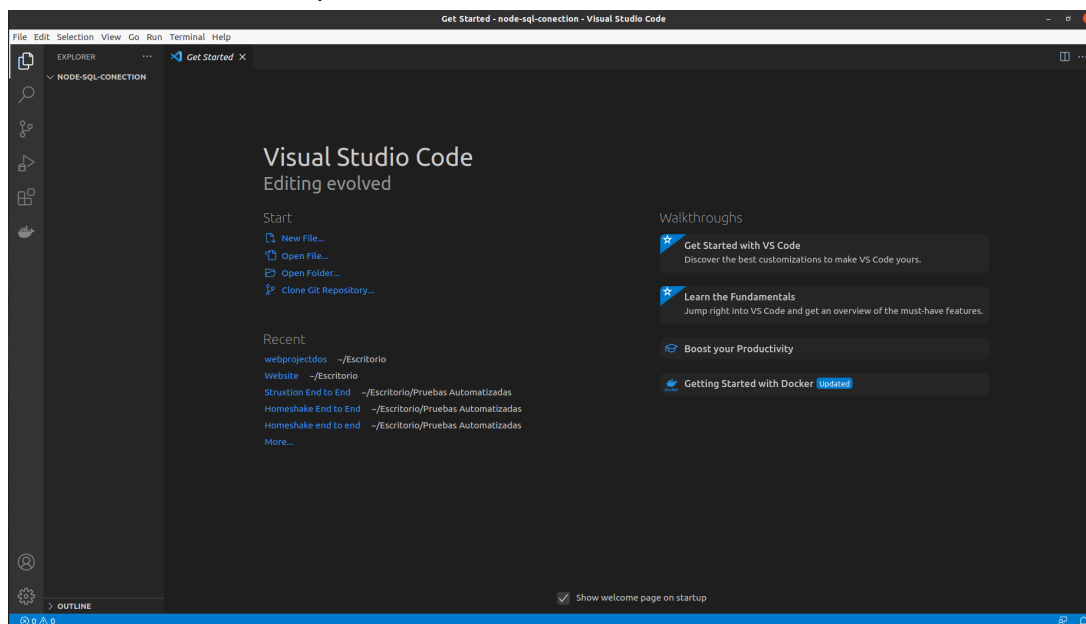
Docker, Nodejs & Mariadb (SQL desde un contenedor de Docker)

1. Verificar que se tenga instalado node y docker, para ello ejecutamos en la terminal los siguientes comandos: `docker --version` y `node --version`.



```
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$ docker --version  
Docker version 20.10.13, build a224086  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$ node --version  
v16.11.0  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$
```

2. Crear una nueva carpeta llamada node-sql-conection
3. Abrir nuestra carpeta en el editor de VisualStudio Code

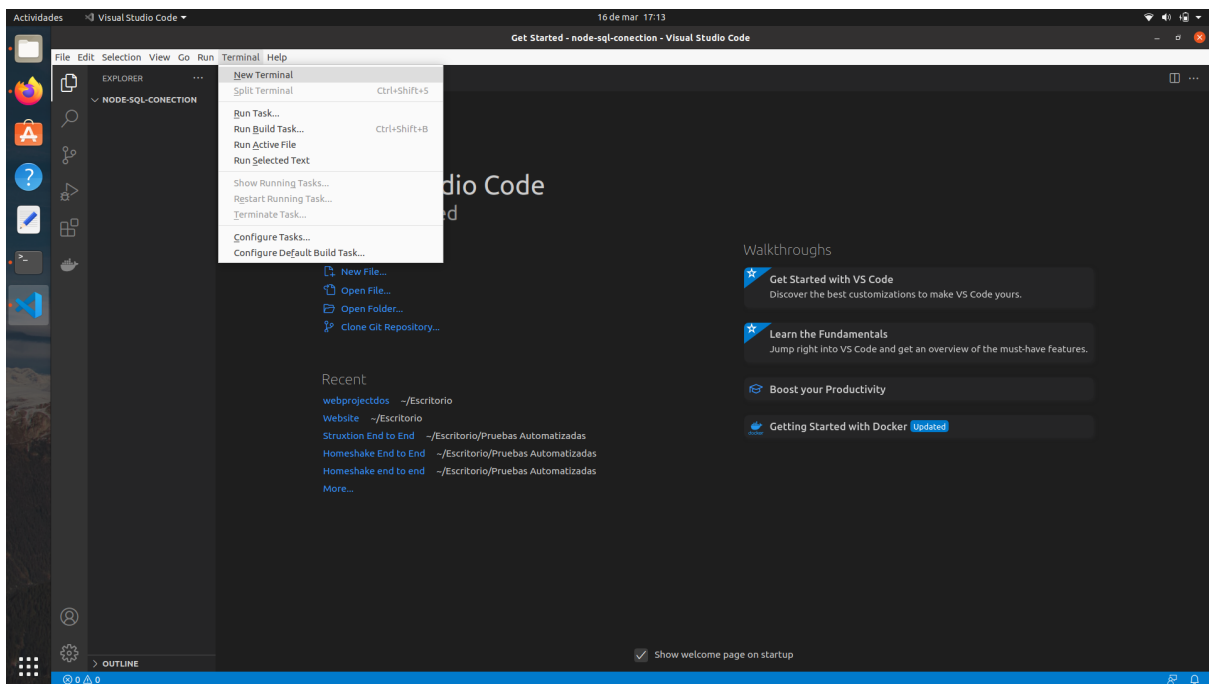


4. Creación del proyecto de node: para ello abrimos una nueva terminal de VisualStudio Code y ejecutamos el comando `npm init` con el fin de inicializar el proyecto, creando así un archivo llamado **package.json**. Dentro de ese archivo, encontrarás metadatos específicos para el proyecto.

Los datos que pedirá son:

- a) El nombre del paquete
- b) Versión
- c) Descripción
- d) El punto de entrada
- e) Comandos de prueba
- f) Repositorio de git
- g) Palabras Clave
- h) Autor
- i) Licencia

Por último, en la confirmación es necesario dar enter para que el proceso finalice.



The image shows two screenshots. The top screenshot is a terminal window with the following text:

```
harvey@harvey-HP-Pavilion-Laptop-15-cw0051a:~/Escritorio/node-sql-conection$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (sql-conection) conexionbdnode
version: (1.0.0) 0.0.1
description: Proyecto con conexion a bd
entry point: (index.js)
test command:
git repository:
keywords: node bd docker test
author: Nicolas Echavarria
license: (ISC)
About to write to /home/harvey/Escritorio/node-sql-conection/package.json:

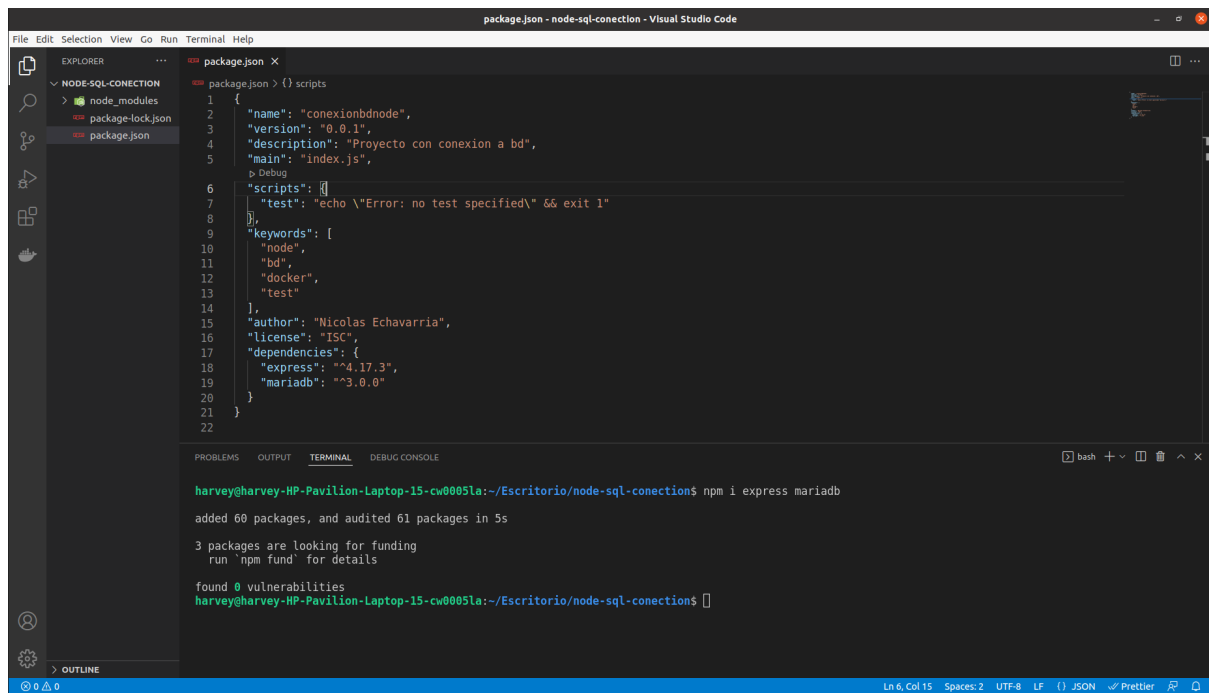
{
  "name": "conexionbdnode",
  "version": "0.0.1",
  "description": "Proyecto con conexion a bd",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "node",
    "bd",
    "docker",
    "test"
  ],
  "author": "Nicolas Echavarria",
  "license": "ISC"
}

Is this OK? (yes)
```

The bottom screenshot is a Visual Studio Code editor window titled "package.json - node-sql-conection - Visual Studio Code". The Explorer sidebar on the left shows the file structure with "package.json" selected. The main editor area displays the content of "package.json" with line numbers 1 through 18:

```
1 {
2   "name": "conexionbdnode",
3   "version": "0.0.1",
4   "description": "Proyecto con conexion a bd",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "keywords": [
10    "node",
11    "bd",
12    "docker",
13    "test"
14  ],
15  "author": "Nicolas Echavarria",
16  "license": "ISC"
17 }
18 |
```

5. Instalar los paquetes necesarios por medio de los comandos: npm i express (framework de node para crear el servidor) y npm i mariadb (drive para base de datos de mariadb - permite utilizar promesas y métodos asíncronos)



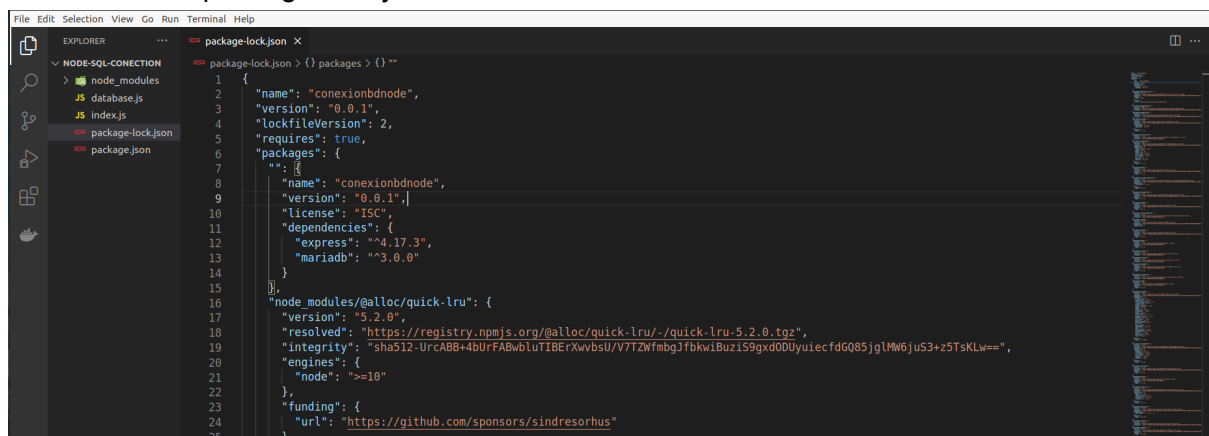
The screenshot shows the Visual Studio Code interface with the `package.json` file open in the editor. The file contains the following JSON:

```
{
  "name": "conexionbdnode",
  "version": "0.0.1",
  "description": "Proyecto con conexion a bd",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [
    "node",
    "bd",
    "docker",
    "test"
  ],
  "author": "Nicolas Echavarria",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.3",
    "mariadb": "^3.0.0"
  }
}
```

The terminal at the bottom shows the output of the command `npm i express mariadb`:

```
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~/Escritorio/node-sql-conexion$ npm i express mariadb
added 60 packages, and audited 61 packages in 5s
3 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~/Escritorio/node-sql-conexion$
```

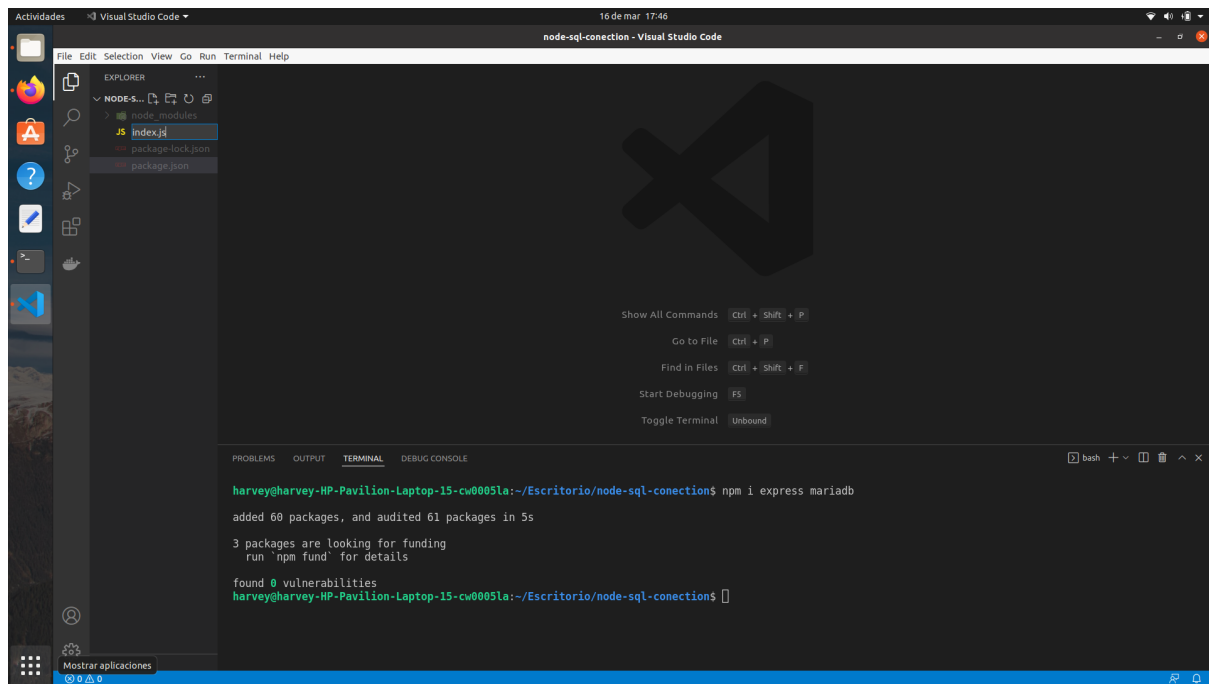
- Una vez ejecutados los comandos se crea una carpeta llamada `node_modules` y el archivo `package-lock.json`



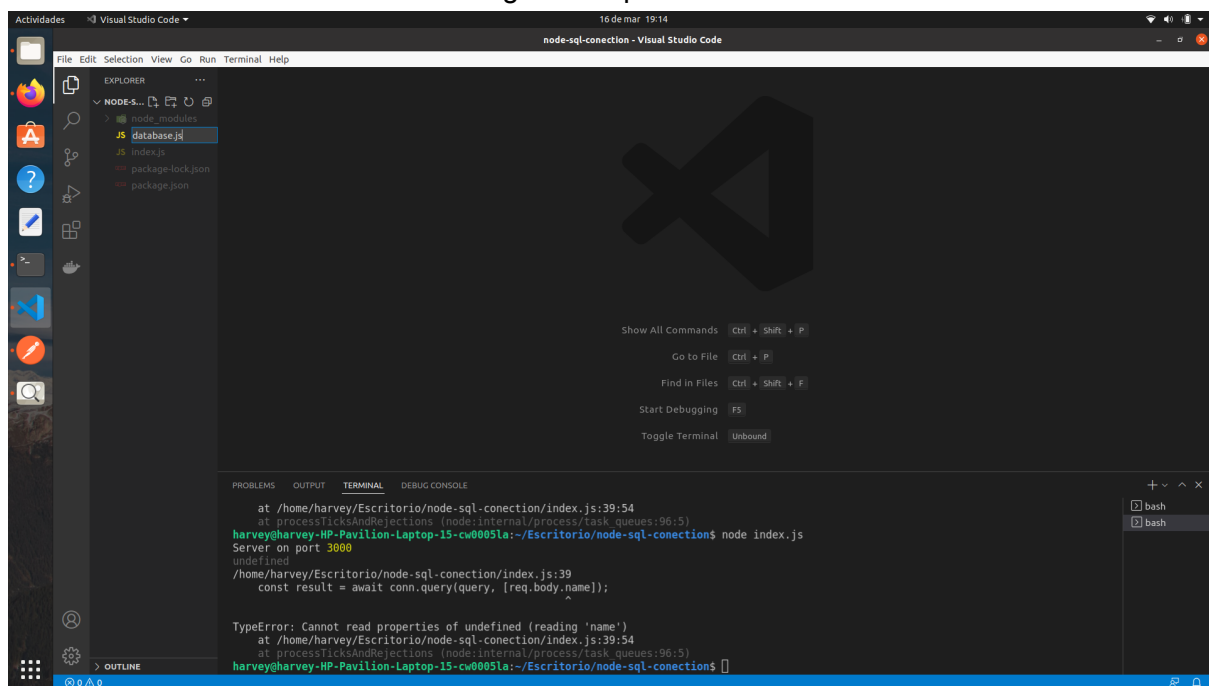
The screenshot shows the Visual Studio Code interface with the `package-lock.json` file open in the editor. The file contains the following JSON:

```
{
  "name": "conexionbdnode",
  "version": "0.0.1",
  "lockfileVersion": 2,
  "requires": true,
  "packages": {
    "": {
      "name": "conexionbdnode",
      "version": "0.0.1",
      "license": "ISC",
      "dependencies": {
        "express": "^4.17.3",
        "mariadb": "^3.0.0"
      }
    },
    "node_modules/@alloc/quick-lru": {
      "version": "5.2.0",
      "resolved": "https://registry.npmjs.org/@alloc/quick-lru/-/quick-lru-5.2.0.tgz",
      "integrity": "sha512-UrkFWB7wJ4uH1vDmCmpSiKqJ0ZdWu6r1Y4ZSOGzuuG1VwX65wJ4s7Z0Z7IuLlK4w4u1K5kY18w8g==",
      "engines": {
        "node": ">=10"
      },
      "funding": {
        "url": "https://github.com/sponsors/sindresorhus"
      }
    }
  }
}
```

- Dentro de la carpeta principal del proyecto creamos un archivo llamado `index.js` con el fin de crear nuestro primer servidor.



8. Dentro de la carpeta principal del proyecto creamos un archivo llamado database.js con el fin de establecer la configuración para nuestra base de datos de mariadb



9. Ejecutamos el siguiente comando: `docker run -p 3307:3306 -d --name mariadb -e MYSQL_ROOT_PASSWORD=mypassword mariadb/server:10.4`, con el fin de descargar la imagen de maria db de docker aplicando variables de entorno y asignación de puertos, la imagen se puede encontrar en el siguiente enlace: https://hub.docker.com/_/mariadb.

```
harvey@harvey-HP-Pavillon-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavillon-L... x harvey@harvey-HP-Pavillon-L... x harvey@harvey-HP-Pavillon-La... x  
REPOSITORY TAG IMAGE ID CREATED SIZE  
dpkg/pgadmin4 latest 4b5bbddb3624 2 days ago 340MB  
postgres latest d7337c283715 8 days ago 376MB  
harvey@harvey-HP-Pavillon-Laptop-15-cw0005la:~$ docker run -p 3307:3306 -d --name mariadb -e MYSQL_ROOT  
T_PASSWORD=mypassword mariadb/server:10.4  
Unable to find image 'mariadb/server:10.4' locally  
10.4: Pulling from mariadb/server  
01bf7da0a88c: Pull complete  
f3b4a5f15c7a: Pull complete  
57ffbe87baa1: Pull complete  
c8e5bbbed1e08: Pull complete  
be83a850f3b2: Pull complete  
80fdb284b6ed: Pull complete  
25f58404dd9a: Pull complete  
8662306d01c4: Pull complete  
5c4bf75b2ed6: Pull complete  
df7572167949: Pull complete  
b3c8322dfdb3: Pull complete  
dbbf63522cd6: Pull complete  
ad5db97f4601: Pull complete  
Digest: sha256:67e523b2e4e627d1fc41c04f1059f955caa97b05cfccaf79ac2ee9669c6e76  
Status: Downloaded newer image for mariadb/server:10.4  
10c3c88b127e5dee80e1d5de548edd02224d9c04fd4d779924ca2e939cd32bcf  
harvey@harvey-HP-Pavillon-Laptop-15-cw0005la:~$
```

10. Ejecutamos docker images para validar que la imagen se descargó.

```
harvey@harvey-HP-Pavillon-Laptop-15-cw0005la:~$ docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
dpkg/pgadmin4 latest 4b5bbddb3624 2 days ago 340MB  
postgres latest d7337c283715 8 days ago 376MB  
nginx latest c919045c4c2b 2 weeks ago 142MB  
mysql/mysql-server latest 434c35b82b08 8 weeks ago 417MB  
mariadb/server 10.4 6bc393df66bb 10 months ago 353MB  
harvey@harvey-HP-Pavillon-Laptop-15-cw0005la:~$
```

11. Ejecutamos docker ps para validar que el contenedor o proceso esté corriendo.

```
harvey@harvey-HP-Pavillon-Laptop-15-cw0005la:~$ docker ps  
CONTAINER ID IMAGE COMMAND NAMES CREATED STATUS  
PORTS  
10c3c88b127e mariadb/server:10.4 "docker-entrypoint.s..." About a minute ago Up About a minute  
0.0.0.0:3307->3306/tcp, :::3307->3306/tcp mariadb
```

12. Ejecutamos el siguiente comando para validar que tenemos acceso a mariadb:
mysql --host 127.0.0.1 -P 3307 -u root -p, la contraseña es aquella dada como
parámetro en el paso 9 (mypassword).

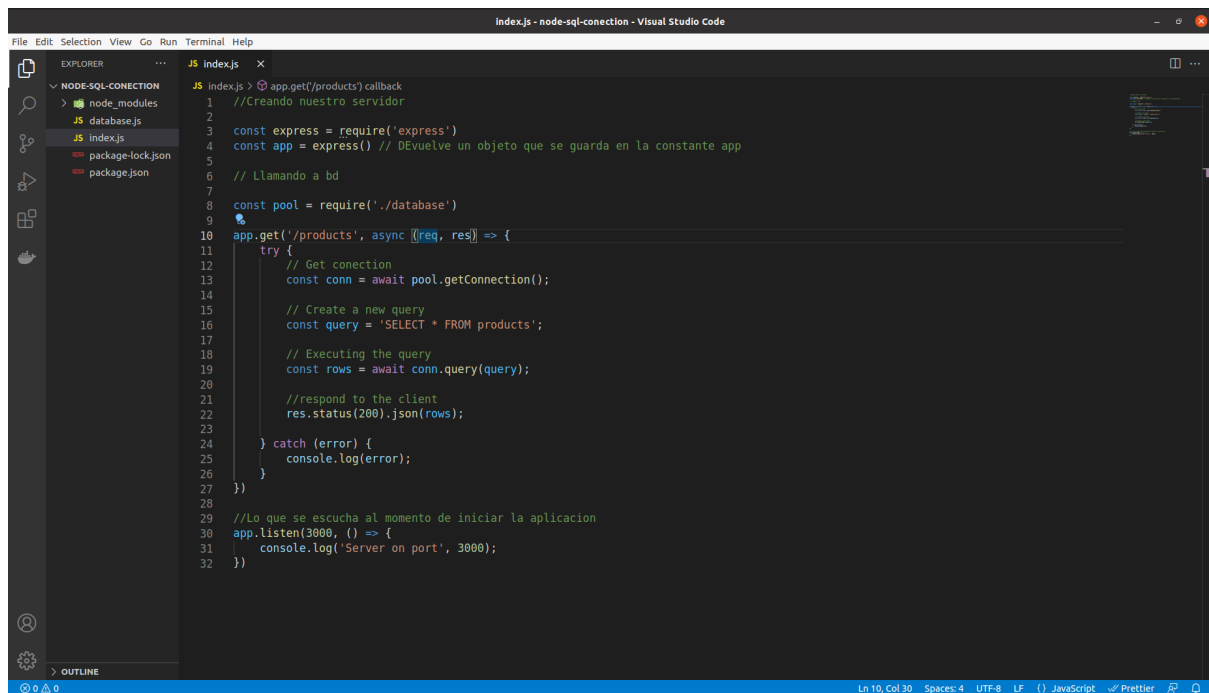
```
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavilion-L... x harvey@harvey-HP-Pavilion-L... x harvey@harvey-HP-Pavilion-La... x  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$ docker ps  
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS  
10c3c88b127e   mariadb/server:10.4                "docker-entrypoint.s..." 6 minutes ago  Up 6 minutes  0.0.0.0:3307->3306/tcp, :::3307->3306/tcp  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$ mysql --host 127.0.0.1 -P 3307 -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 8  
Server version: 10.4.19-MariaDB-1:10.4.19+maria-bionic mariadb.org binary distribution  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]>
```

13. Ejecutamos las siguientes sentencias SQL:

```
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$ mysql --host 127.0.0.1 -P 3307 -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 8  
Server version: 10.4.19-MariaDB-1:10.4.19+maria-bionic mariadb.org binary distribution  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
+-----+  
3 rows in set (0.001 sec)  
MariaDB [(none)]> CREATE DATABASE mydatabase;  
Query OK, 1 row affected (0.001 sec)  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mydatabase |  
| mysql |  
| performance_schema |  
+-----+  
4 rows in set (0.001 sec)  
MariaDB [(none)]> USE mydatabase;  
Database changed  
MariaDB [mydatabase]> SELECT DATABASE();  
+-----+  
| DATABASE() |  
+-----+  
| mydatabase |  
+-----+  
1 row in set (0.000 sec)  
MariaDB [mydatabase]> CREATE TABLE products(name VARCHAR(100));  
Query OK, 0 rows affected (0.129 sec)
```

```
MariaDB [mydatabase]> DESCRIBE products;  
+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+  
| name | varchar(100) | YES | | NULL | |  
+-----+  
1 row in set (0.001 sec)  
MariaDB [mydatabase]> INSERT INTO products VALUES ('laptop'), ('mouse'), ('keyboard');  
Query OK, 3 rows affected (0.022 sec)  
Records: 3 Duplicates: 0 Warnings: 0  
MariaDB [mydatabase]> SELECT * FROM products;  
+-----+  
| name |  
+-----+  
| laptop |  
| mouse |  
| keyboard |  
+-----+  
3 rows in set (0.001 sec)
```

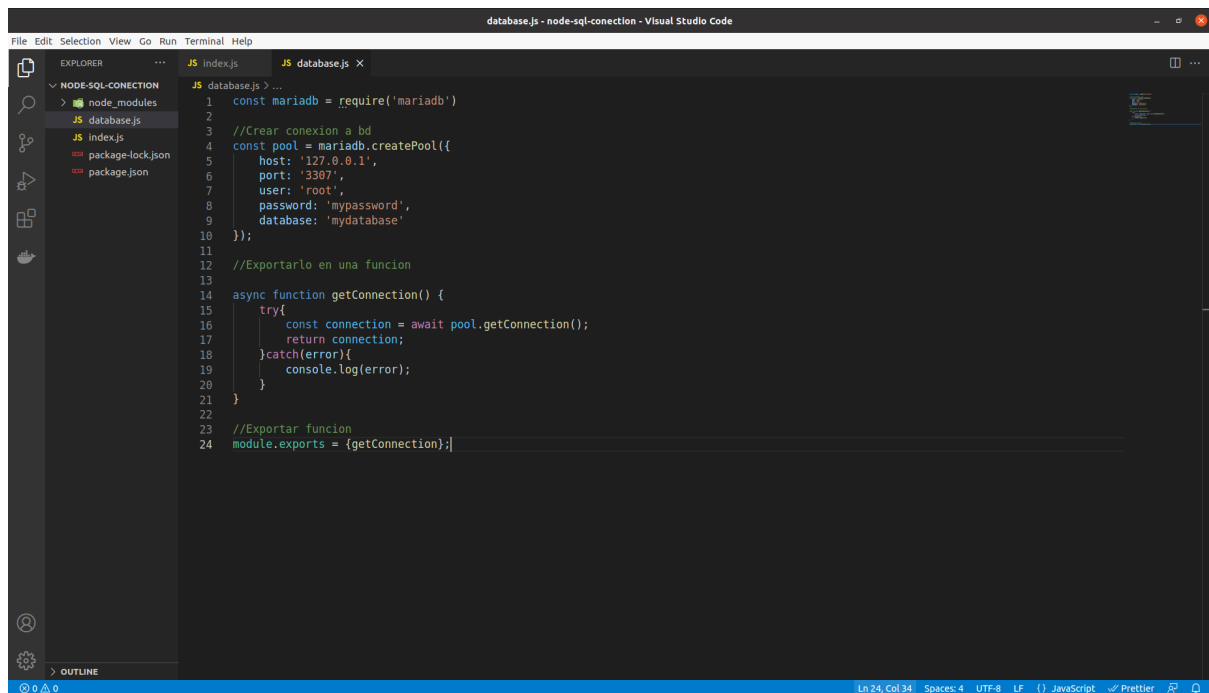
14. Agregar el siguiente código en los archivos index.js y database.js



```
Index.js - node-sql-conection - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
  NODE-SQL-CONNECTION
    node_modules
    JS database.js
    JS index.js
    package-lock.json
    package.json

JS index.js
1 //Creando nuestro servidor
2
3 const express = require('express')
4 const app = express() // Devuelve un objeto que se guarda en la constante app
5
6 // Llamando a bd
7
8 const pool = require('./database')
9
10 app.get('/products', async (req, res) => {
11   try {
12     // Get connection
13     const conn = await pool.getConnection();
14
15     // Create a new query
16     const query = 'SELECT * FROM products';
17
18     // Executing the query
19     const rows = await conn.query(query);
20
21     //respond to the client
22     res.status(200).json(rows);
23   } catch (error) {
24     console.log(error);
25   }
26 }
27
28 //Lo que se escucha al momento de iniciar la aplicacion
29
30 app.listen(3000, () => {
31   console.log('Server on port', 3000);
32 })
```

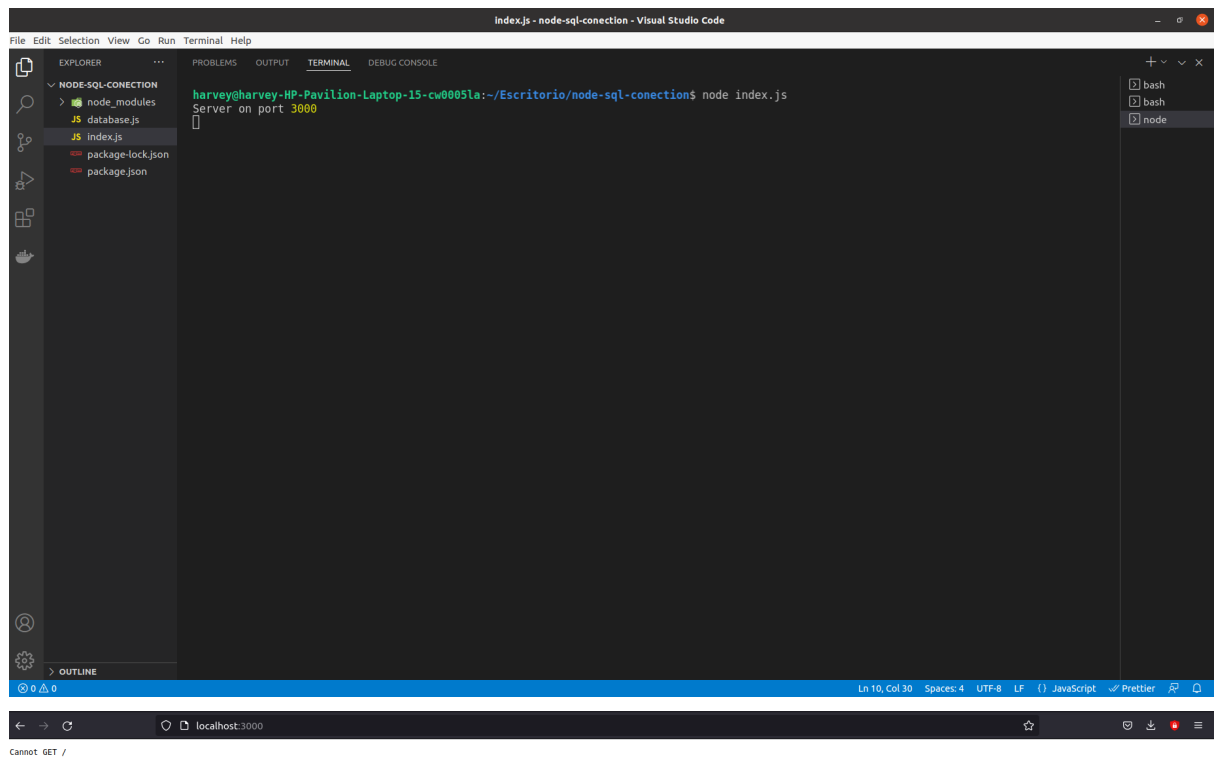


```
database.js - node-sql-conection - Visual Studio Code
File Edit Selection View Go Run Terminal Help

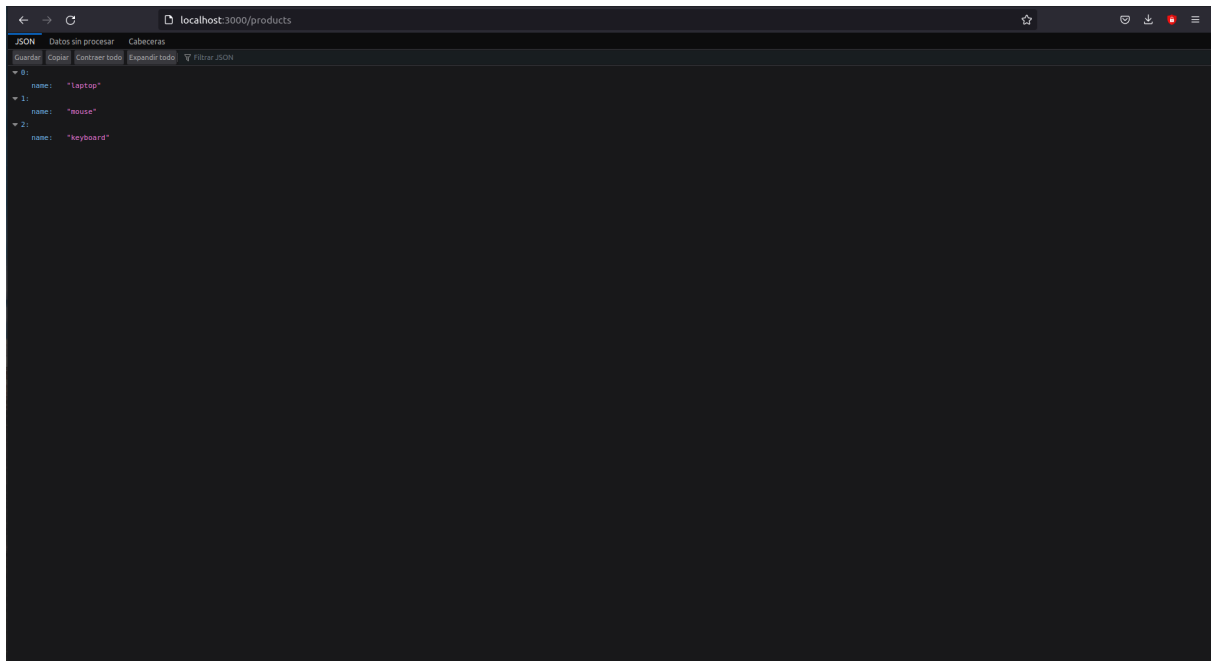
EXPLORER
  NODE-SQL-CONNECTION
    node_modules
    JS database.js
    JS index.js
    package-lock.json
    package.json

JS database.js
1 const mariadb = require('mariadb')
2
3 //Crear conexion a bd
4 const pool = mariadb.createPool({
5   host: '127.0.0.1',
6   port: '3307',
7   user: 'root',
8   password: 'mypassword',
9   database: 'mydatabase'
10 });
11
12 //Exportarlo en una funcion
13
14 async function getConnection() {
15   try{
16     const connection = await pool.getConnection();
17     return connection;
18   }catch(error){
19     console.log(error);
20   }
21 }
22
23 //Exportar funcion
24 module.exports = {getConnection};
```

15. Ejecutamos en la terminal el siguiente comando: `node index.js` y verificamos en el navegador accediendo a `localhost:3000`, sin embargo, al no tener manejadores de ruta es normal que se genere el error `Cannot GET /` de la página.



16. Accedemos a la ruta <http://localhost:3000/products> para poder visualizar la consulta realizada.



17. Realizamos las pruebas con postman o insomnia, para ello debemos descargar alguno de los dos programas, y realizar los siguientes pasos:

