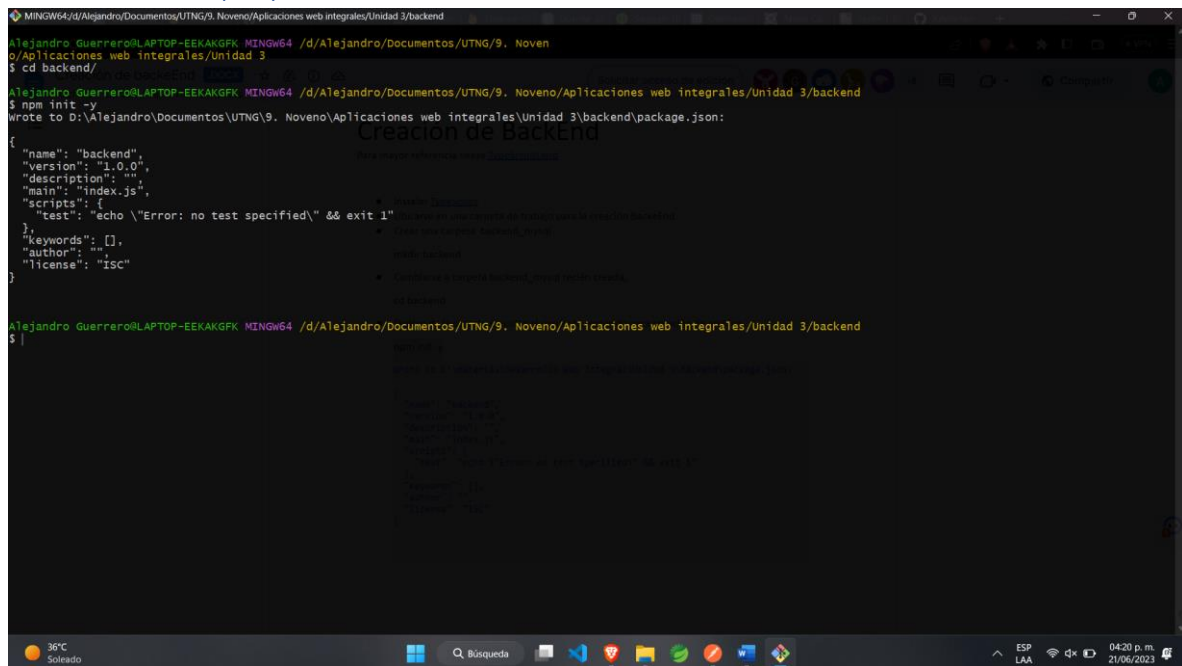


Inicializando el proyecto.

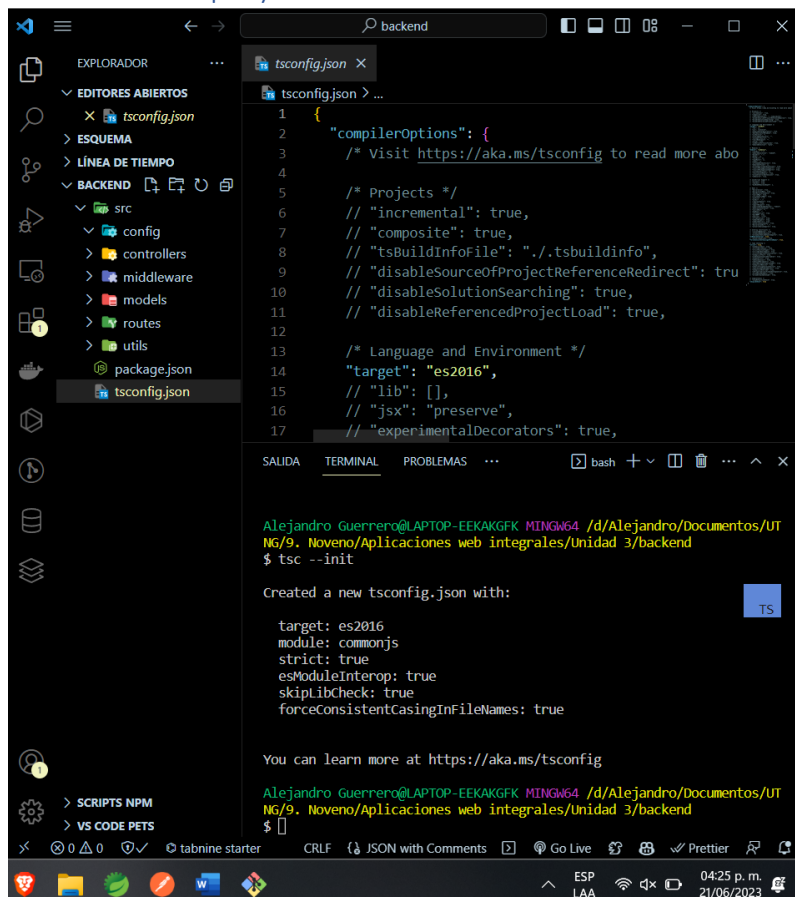


```
MINGW64 /d/Alejandro/Documentos/UTNG/9. Noveno/Aplicaciones web integrales/Unidad 3/backend
Alejandro Guerrero@LAPTOP-EEKAKGFK MINGW64 /d/Alejandro/Documentos/UTNG/9. Noveno/Aplicaciones web integrales/Unidad 3/backend
$ npm init -y
Wrote to D:\Alejandro\Documentos\UTNG\9. Noveno\Aplicaciones web integrales\Unidad 3\backend\package.json:

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

Alejandro Guerrero@LAPTOP-EEKAKGFK MINGW64 /d/Alejandro/Documentos/UTNG/9. Noveno/Aplicaciones web integrales/Unidad 3/backend
$
```

Estructura del proyecto.



```
tsconfig.json
1 {
2   "compilerOptions": {
3     /* Visit https://aka.ms/tsconfig to read more about this file */
4
5     /* Projects */
6     // "incremental": true,
7     // "composite": true,
8     // "tsBuildInfoFile": "./.tsbuildinfo",
9     // "disableSourceOfProjectReferenceRedirect": true,
10    // "disableSolutionSearching": true,
11    // "disableReferencedProjectLoad": true,
12
13    /* Language and Environment */
14    "target": "es2016",
15    // "lib": [],
16    // "jsx": "preserve",
17    // "experimentalDecorators": true,
```

EXPLORADOR

- EDITORIOS ABIERTOS
 - tsconfig.json
- ESQUEMA
- LÍNEA DE TIEMPO
- BACKEND
 - src
 - config
 - controllers
 - middleware
 - models
 - routes
 - utils
 - package.json
 - tsconfig.json

SCRIPTS NPM

VS CODE PETS

terminal output:

```
Alejandro Guerrero@LAPTOP-EEKAKGFK MINGW64 /d/Alejandro/Documentos/UTNG/9. Noveno/Aplicaciones web integrales/Unidad 3/backend
$ tsc --init

Created a new tsconfig.json with:

target: es2016
module: commonjs
strict: true
esModuleInterop: true
skipLibCheck: true
forceConsistentCasingInFileNames: true

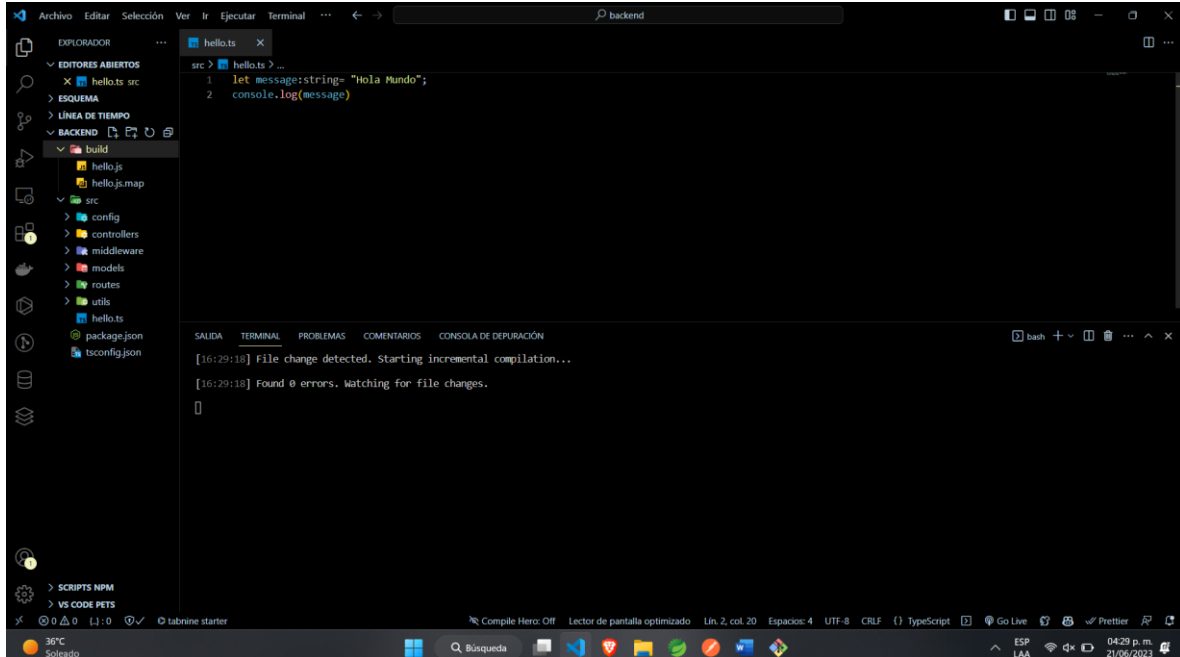
You can learn more at https://aka.ms/tsconfig

Alejandro Guerrero@LAPTOP-EEKAKGFK MINGW64 /d/Alejandro/Documentos/UTNG/9. Noveno/Aplicaciones web integrales/Unidad 3/backend
$
```

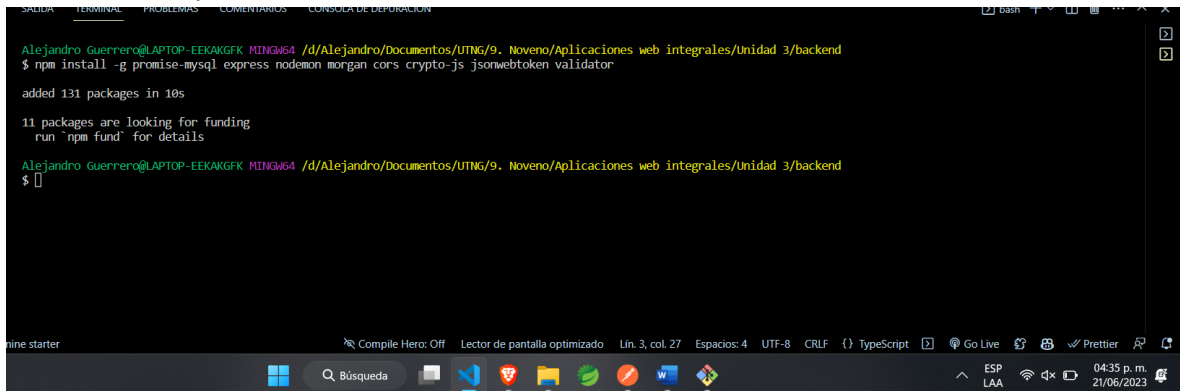
Configurando el proyecto.

```
53 // "declarationMap": true,           /* Create sourcemaps for d.ts files. */
54 // "emitDeclarationOnly": true,      /* Only output d.ts files and not JavaScript files. */
55 "sourceMap": true,                  /* Create source map files for emitted JavaScript files. */
56 // "inlineSourceMap": true,          /* Include sourcemap files inside the emitted JavaScript. */
57 // "outFile": "./",                  /* Specify a file that bundles all outputs into one JavaScript file. If 'declaration' is true,
58 "outDir": "./build",                 /* Specify an output folder for all emitted files. */
59 // "removeComments": true,           /* Disable emitting comments. */
60 // "noEmit": true,                   /* Disable emitting files from a compilation. */
61 // "importHelpers": true,            /* Allow importing helper functions from tslib once per project, instead of including them per
62 // "importsNotUsedAsValues": "remove", /* Specify emit/checking behavior for imports that are only used for types. */
```

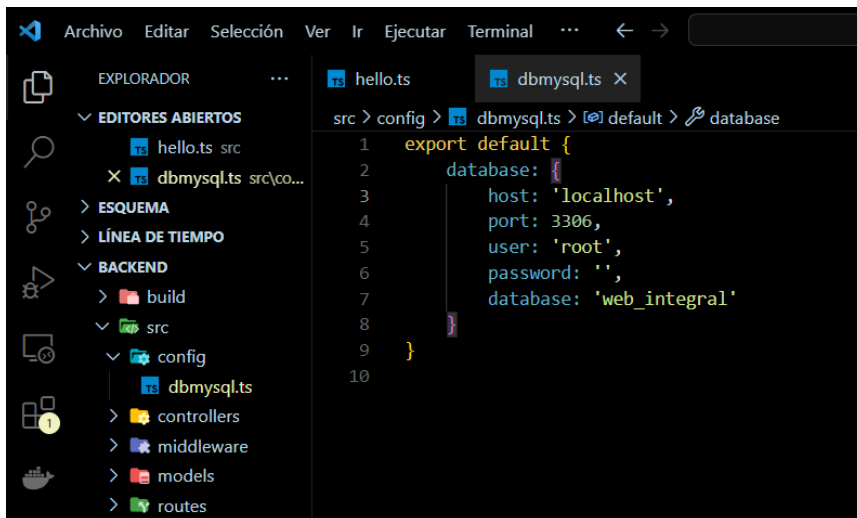
Ejecutando proyecto:



Instalando dependencias.



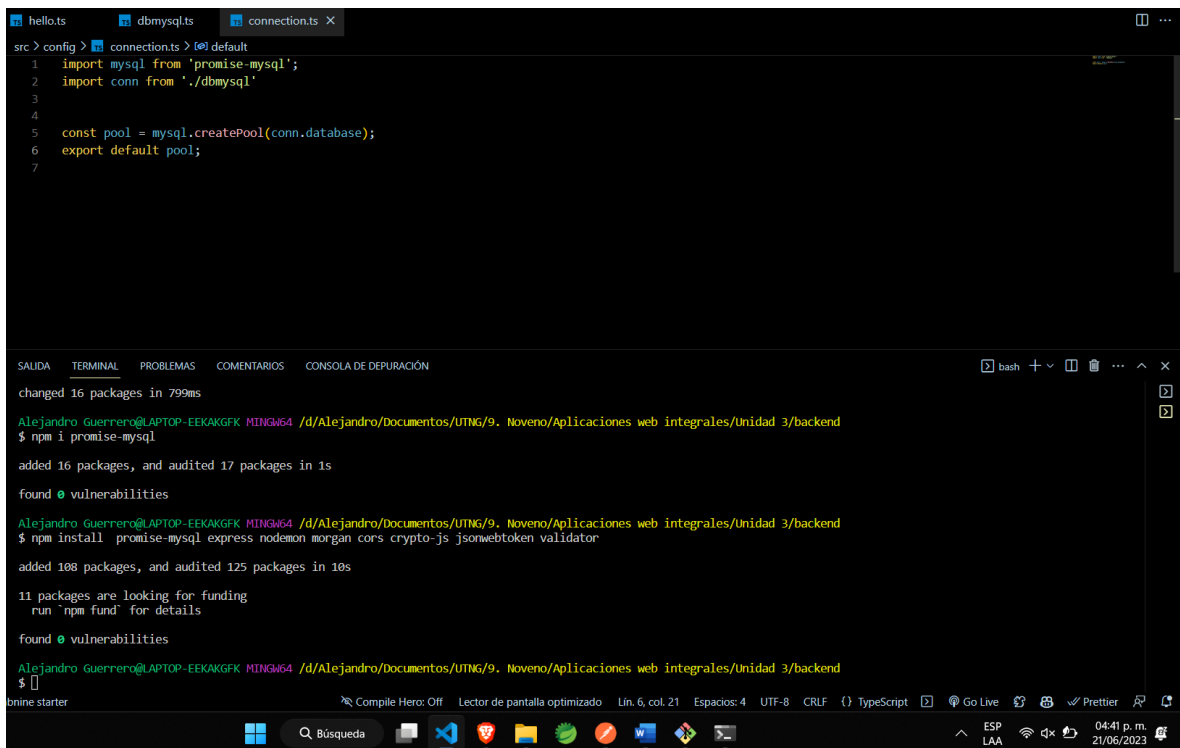
Conexión a la base de datos:



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure: 'EDITORES ABIERTOS' with 'hello.ts src' and 'dbmysql.ts src/co...'; 'ESQUEMA'; 'LÍNEA DE TIEMPO'; 'BACKEND' with 'build', 'src', 'config', 'controllers', 'middleware', 'models', and 'routes'. The 'config' folder is expanded, showing 'dbmysql.ts'. The main editor area shows the 'dbmysql.ts' file with the following code:

```
src > config > dbmysql.ts > default > database
1 export default {
2   database: {
3     host: 'localhost',
4     port: 3306,
5     user: 'root',
6     password: '',
7     database: 'web_integral'
8   }
9 }
10
```

Creando el Pool de conexión.



The screenshot shows the Visual Studio Code interface. The Explorer sidebar shows the project structure with 'connection.ts' selected. The main editor area shows the 'connection.ts' file with the following code:

```
src > config > connection.ts > default
1 import mysql from 'promise-mysql';
2 import conn from './dbmysql'
3
4
5 const pool = mysql.createPool(conn.database);
6 export default pool;
7
```

Below the editor, the Terminal panel shows the output of several npm commands:

```
SALIDA TERMINAL PROBLEMAS COMENTARIOS CONSOLA DE DEPURACIÓN
changed 16 packages in 799ms
Alejandro Guerrero@LAPTOP-EEKAGFK MINGW64 /d/Alejandro/Documentos/UTMG/9. Noveno/Aplicaciones web integrales/Unidad 3/backend
$ npm i promise-mysql
added 16 packages, and audited 17 packages in 1s
found 0 vulnerabilities
Alejandro Guerrero@LAPTOP-EEKAGFK MINGW64 /d/Alejandro/Documentos/UTMG/9. Noveno/Aplicaciones web integrales/Unidad 3/backend
$ npm install promise-mysql express nodemon morgan cors crypto-js jsonwebtoken validator
added 188 packages, and audited 125 packages in 10s
11 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
Alejandro Guerrero@LAPTOP-EEKAGFK MINGW64 /d/Alejandro/Documentos/UTMG/9. Noveno/Aplicaciones web integrales/Unidad 3/backend
$
```

The status bar at the bottom shows 'bnine starter', 'Compile Hero: Off', 'Lector de pantalla optimizado', 'Lin. 6, col. 21', 'Espacios: 4', 'UTF-8', 'CRLF', 'TypeScript', 'Go Live', 'Prettier', and the system clock '04:41 p. m. 21/06/2023'.

Creando la base de datos:

```
MariaDB [web_integral]> create database web_integral
-> ;
Query OK, 1 row affected (0.011 sec)

MariaDB [web_integral]> use web_integral;
Database changed
MariaDB [web_integral]> show databases;
+-----+
| Database |
+-----+
| carreras |
| docfee   |
| escuela  |
| information_schema |
| mccartondb |
| mysql    |
| performance_schema |
| practicasda |
| proveedores |
| sensores |
| sys      |
| web_integral |
| web_integral |
+-----+
13 rows in set (0.001 sec)
```

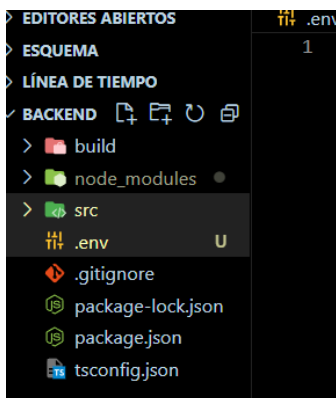
Creando tabla usuario:

```
MariaDB [web_integral]> create table Usuario(
-> id int primary key auto_increment,
-> username varchar(30),
-> password varchar(20),
-> name varchar(80)
-> );
Query OK, 0 rows affected (0.024 sec)

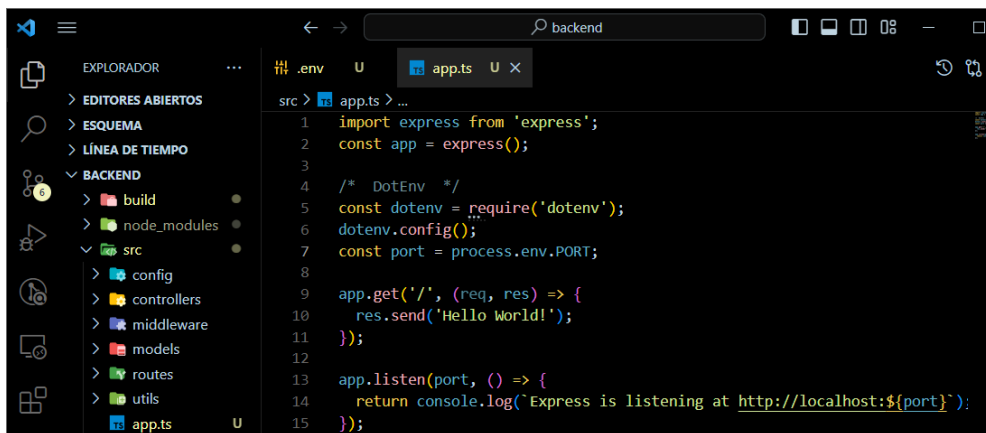
MariaDB [web_integral]> |
```



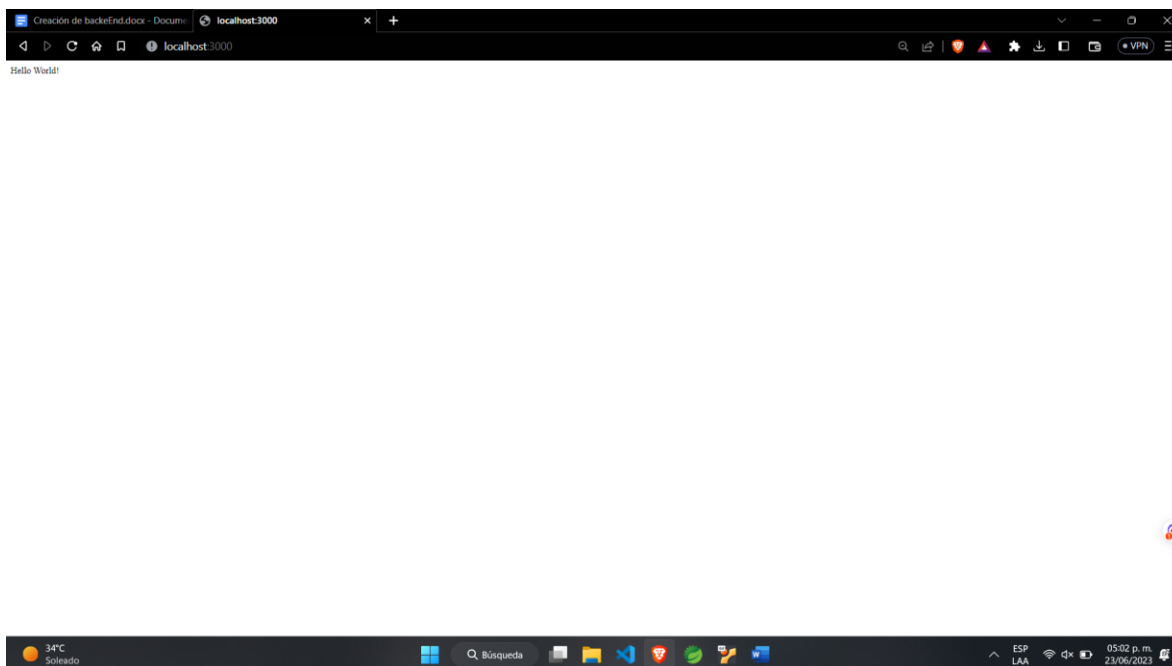
Crear archivo .env



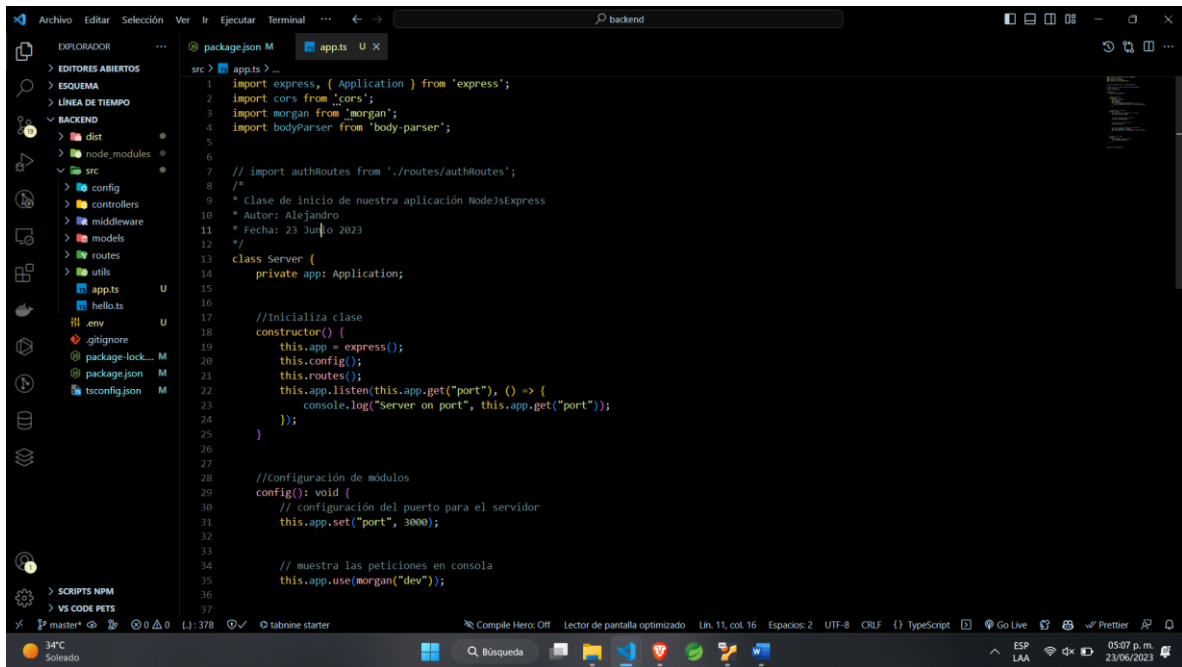
Archivo app.ts



Ejecutando proyecto.

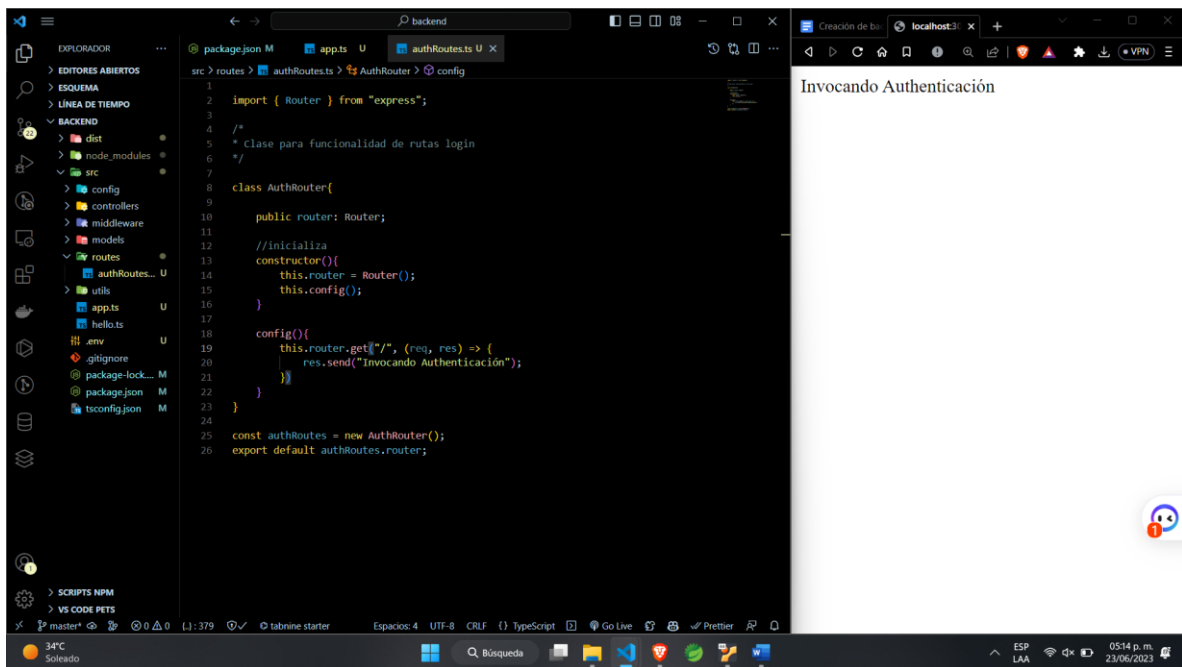


Redefiniendo archivo app.ts



```
1 import express, { Application } from 'express';
2 import cors from 'cors';
3 import morgan from 'morgan';
4 import bodyParser from 'body-parser';
5
6
7 // import authRoutes from './routes/authRoutes';
8 /*
9  * Clase de inicio de nuestra aplicación NodejsExpress
10  * Autor: Alejandro
11  * Fecha: 23 Junio 2023
12  */
13 class Server {
14   private app: Application;
15
16
17   //Inicializa clase
18   constructor() {
19     this.app = express();
20     this.config();
21     this.routes();
22     this.app.listen(this.app.get("port"), () => {
23       console.log("Server on port", this.app.get("port"));
24     });
25   }
26
27
28   //Configuración de módulos
29   config(): void {
30     // configuración del puerto para el servidor
31     this.app.set("port", 3000);
32
33
34     // muestra las peticiones en consola
35     this.app.use(morgan("dev"));
36
37
38   }
39 }
```

Definiendo rutas

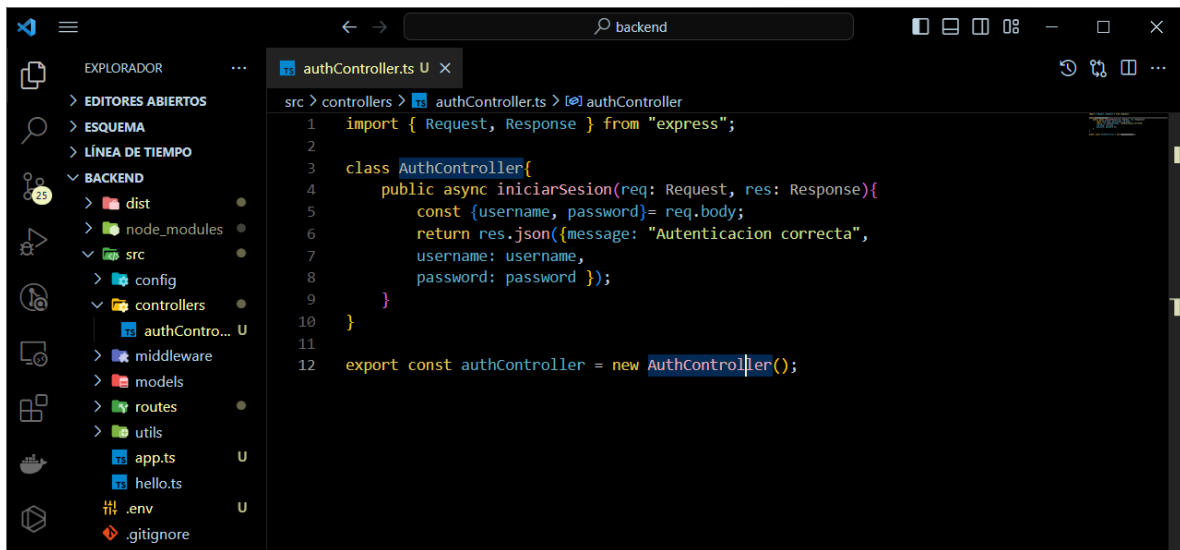


```
1 import { Router } from 'express';
2
3 /*
4  * Clase para funcionalidad de rutas login
5  */
6
7 class AuthRouter {
8   public router: Router;
9
10
11   //Inicializa
12   constructor() {
13     this.router = Router();
14     this.config();
15   }
16
17   config() {
18     this.router.get("/", (req, res) => {
19       res.send("Invocando Autenticación");
20     });
21   }
22 }
23
24 const authRoutes = new AuthRouter();
25 export default authRoutes.router;
```

Creación de bu... localhost:3000

Invocando Autenticación

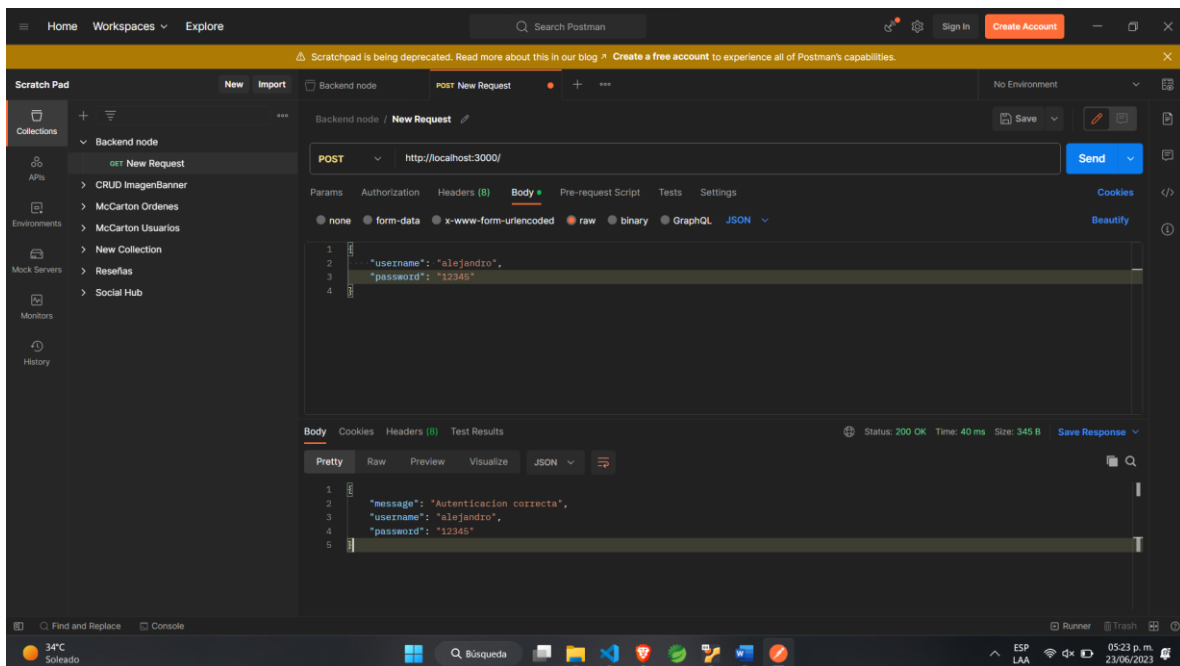
Definiendo controlador



The image shows a Visual Studio Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'dist', 'node_modules', 'src', 'config', 'controllers', 'middleware', 'models', 'routes', 'utils', and files like 'app.ts', 'hello.ts', '.env', and '.gitignore'. The code editor displays the 'authController.ts' file with the following code:

```
src > controllers > authController.ts > [X] authController
1  import { Request, Response } from "express";
2
3  class AuthController{
4      public async iniciarSesion(req: Request, res: Response){
5          const {username, password}= req.body;
6          return res.json({message: "Autenticacion correcta",
7                          username: username,
8                          password: password });
9      }
10 }
11
12 export const authController = new AuthController();
```

Haciendo prueba con postman



Configurar base de datos

Creando tabla:

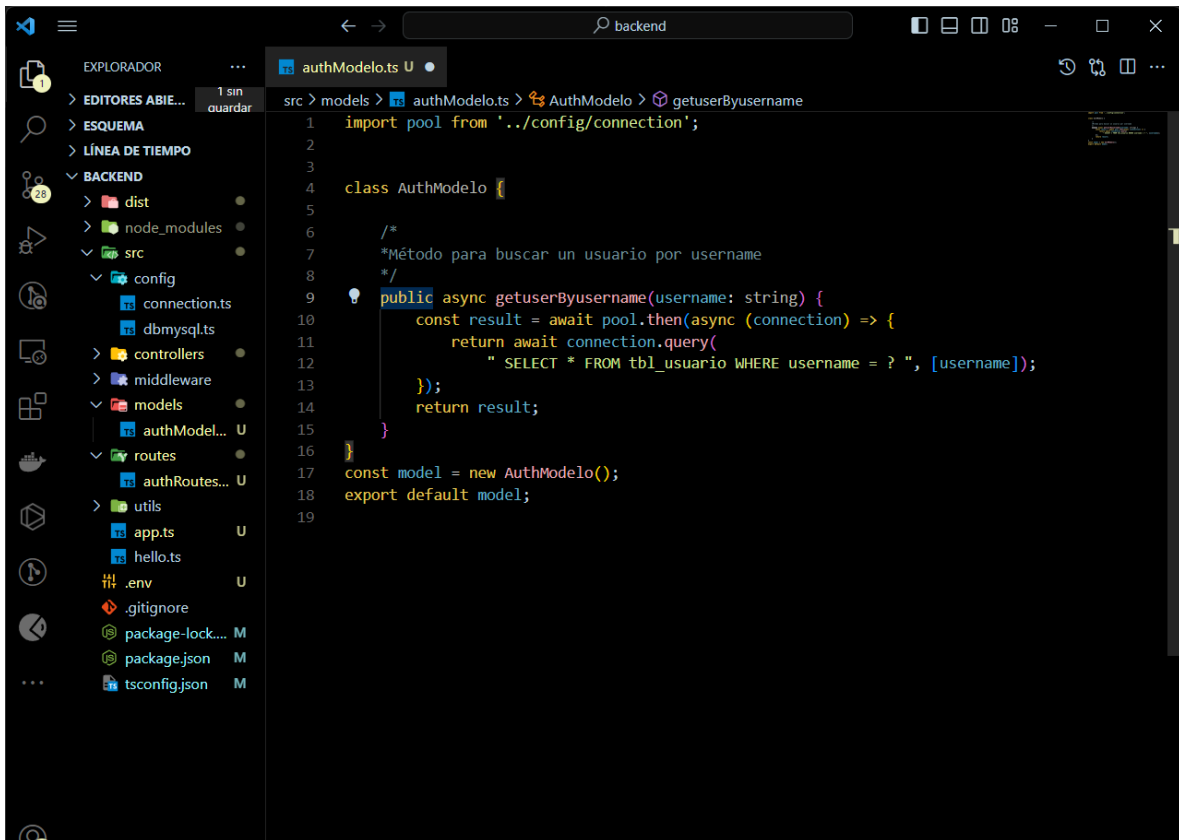
```
MariaDB [web_integral]> CREATE TABLE tbl_usuario (
  ->   username varchar(30) primary key,
  ->   password varchar(250),
  ->   role varchar(20)
  -> );
Query OK, 0 rows affected (0.025 sec)

MariaDB [web_integral]> |
```

```
MariaDB [web_integral]> insert into tbl_usuario(username, password, role) values('progra01','123456','admin');
Query OK, 1 row affected (0.046 sec)

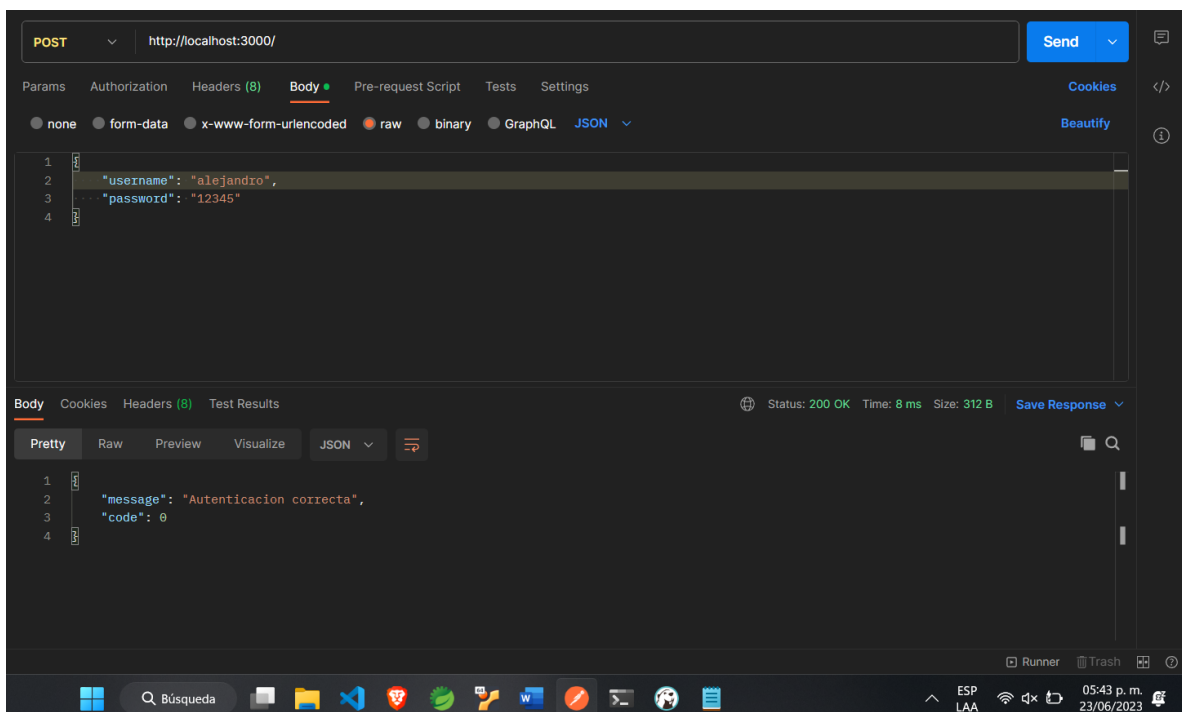
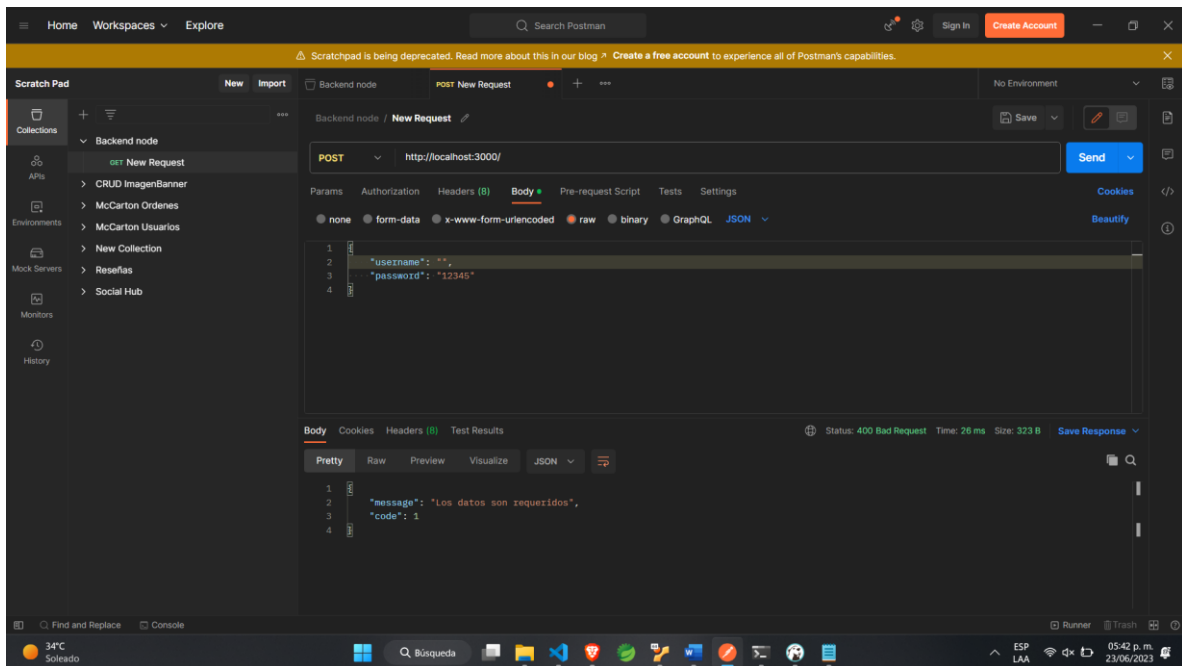
MariaDB [web_integral]> |
```

Modelos

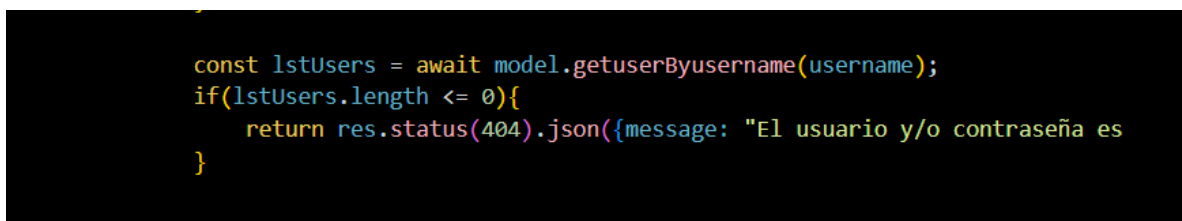


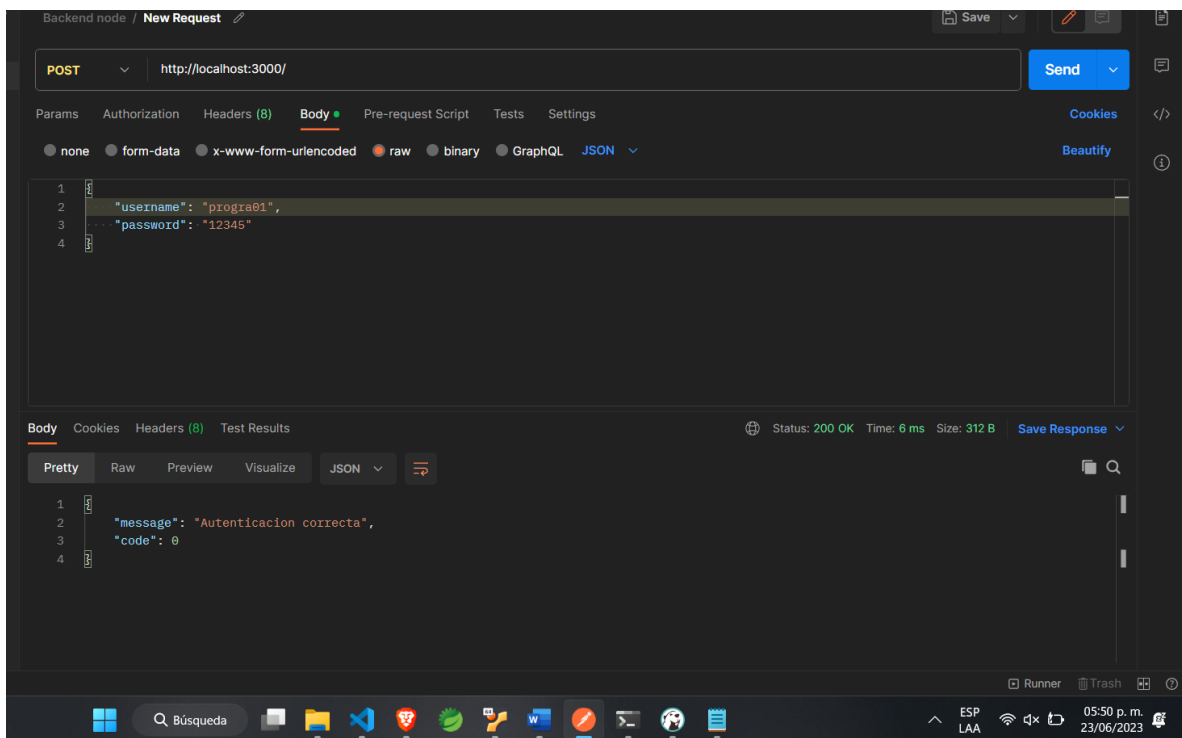
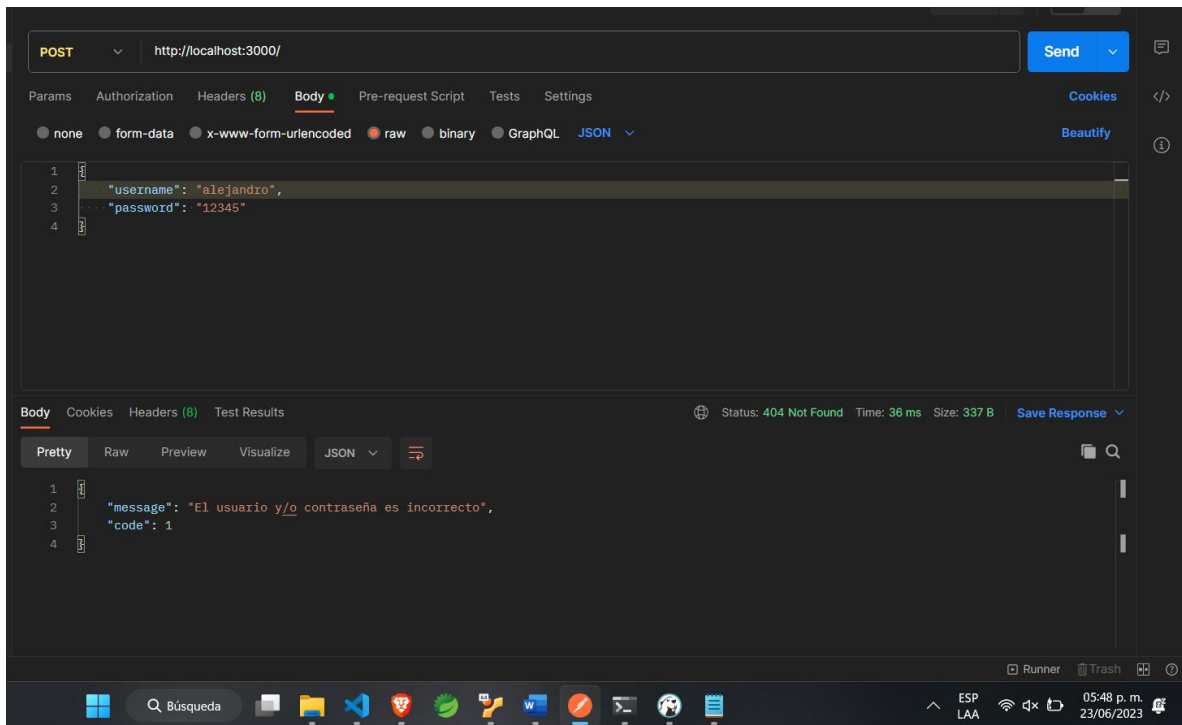
```
src > models > authModelo.ts > AuthModelo > getuserByUsername
1  import pool from '../config/connection';
2
3
4  class AuthModelo {
5
6      /*
7       *Método para buscar un usuario por username
8       */
9      public async getuserByUsername(username: string) {
10         const result = await pool.then(async (connection) => {
11             return await connection.query(
12                 " SELECT * FROM tbl_usuario WHERE username = ? ", [username]);
13             });
14         return result;
15     }
16 }
17 const model = new AuthModelo();
18 export default model;
19
```

Ejecutando petición.



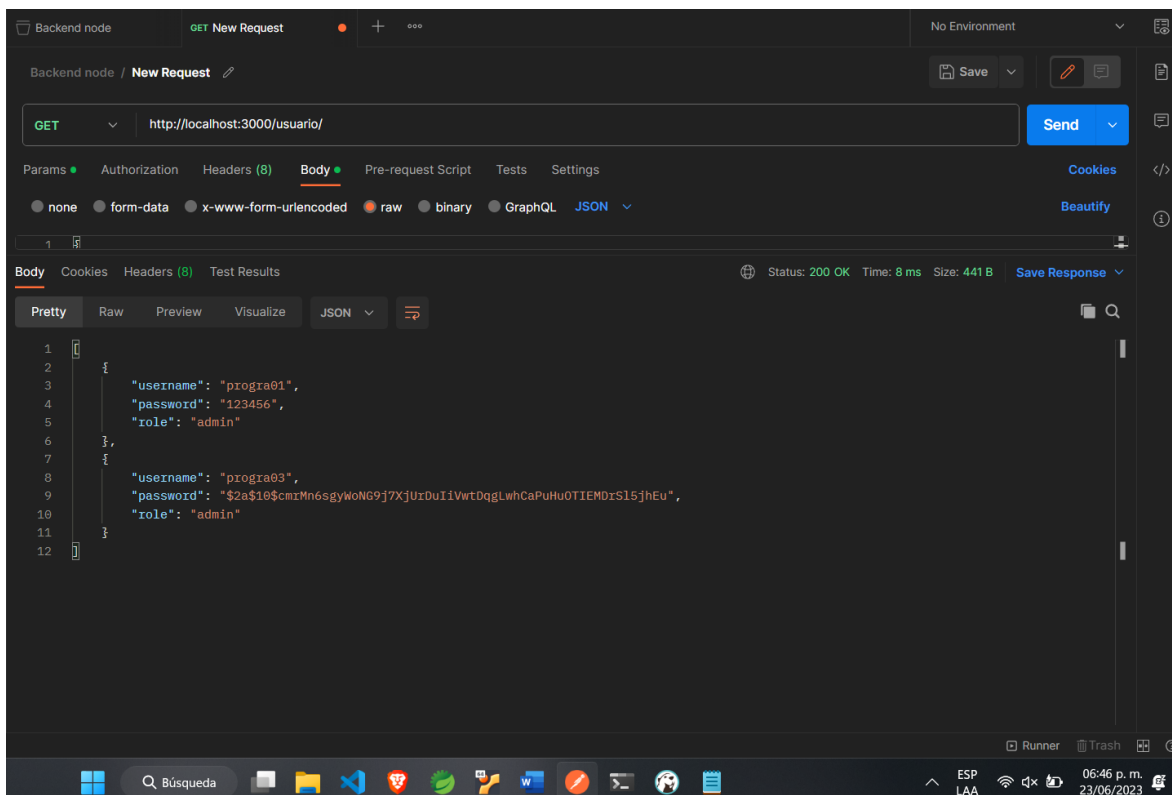
Definiendo el acceso a los datos de datos



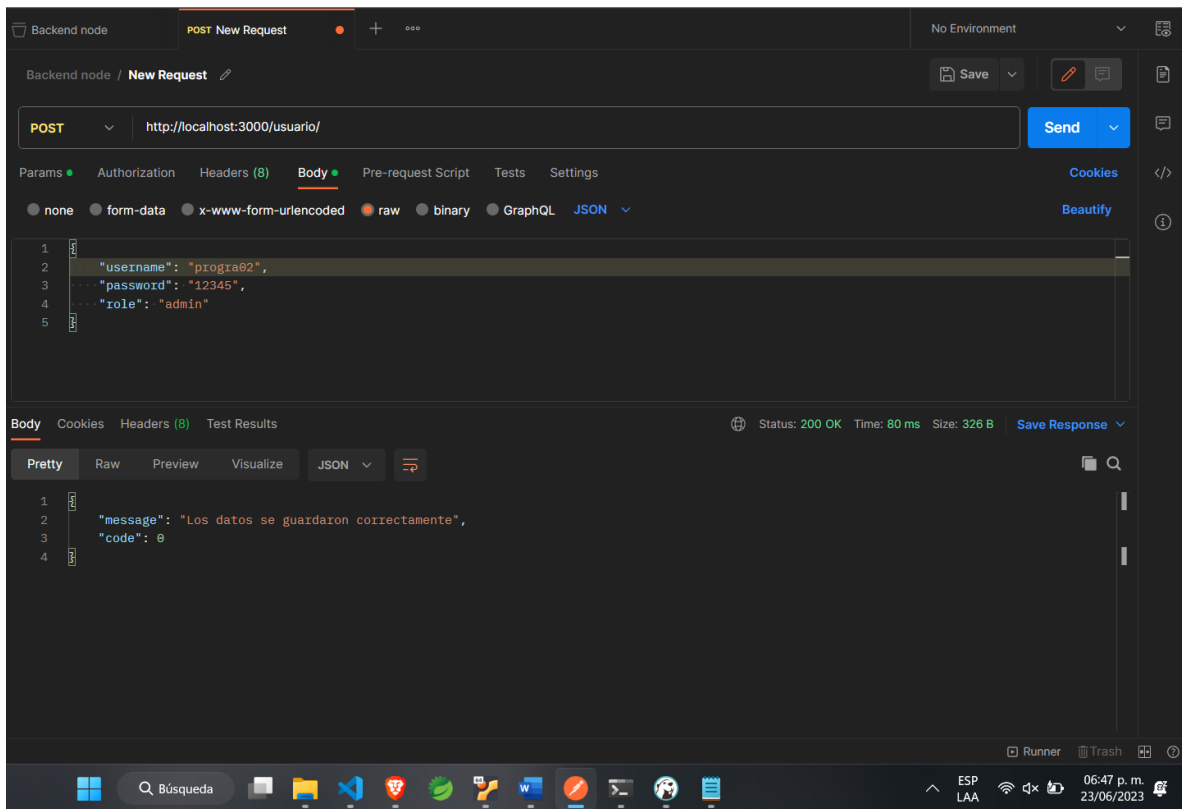


Peticiones:

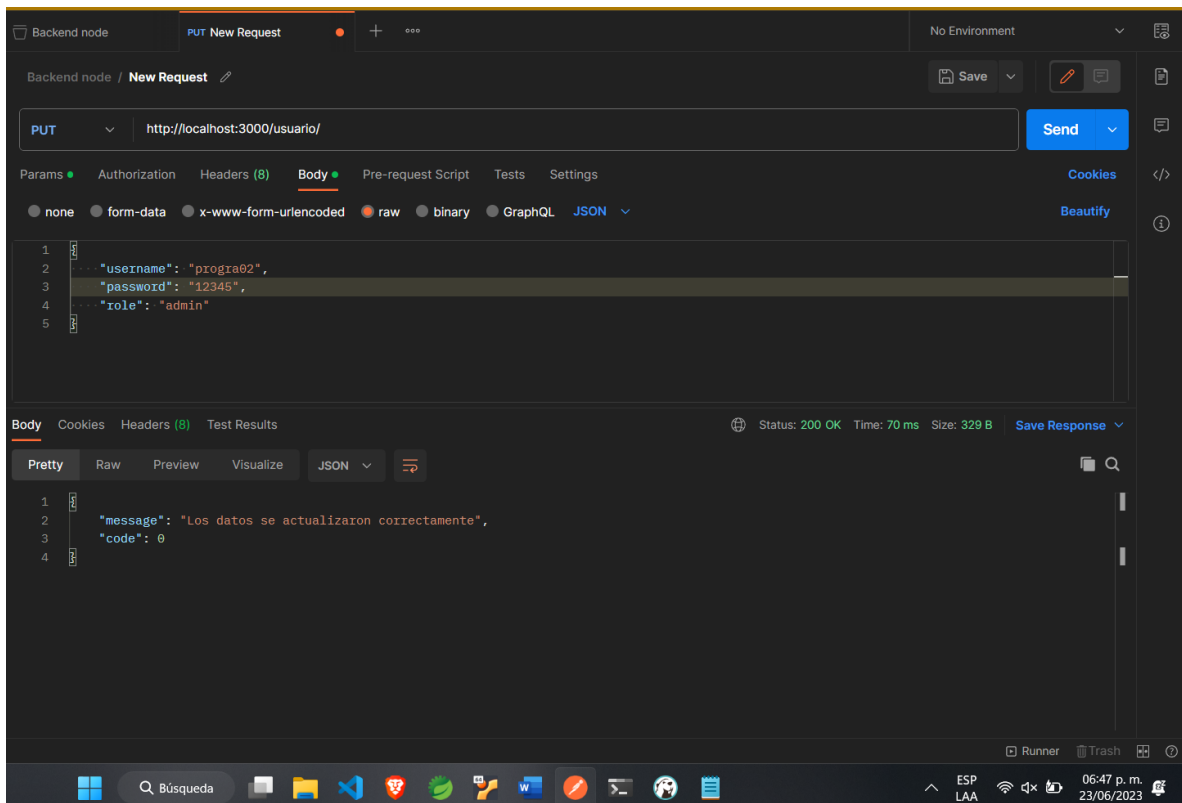
Listar:



Insertar:



Actualizar:



Eliminar:

Backend node / **New Request**

DELETE `http://localhost:3000/usuario/progra02` **Send**

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "username": "progra02",
3   "password": "12345",
4   "role": "admin"
5 }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 7 ms Size: 327 B Save Response

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "message": "Los datos se eliminaron correctamente",
3   "code": 0
4 }
```

Runner Trash

Búsqueda

ESP LAA 06:47 p.m. 23/06/2023

JSON Wen Token.

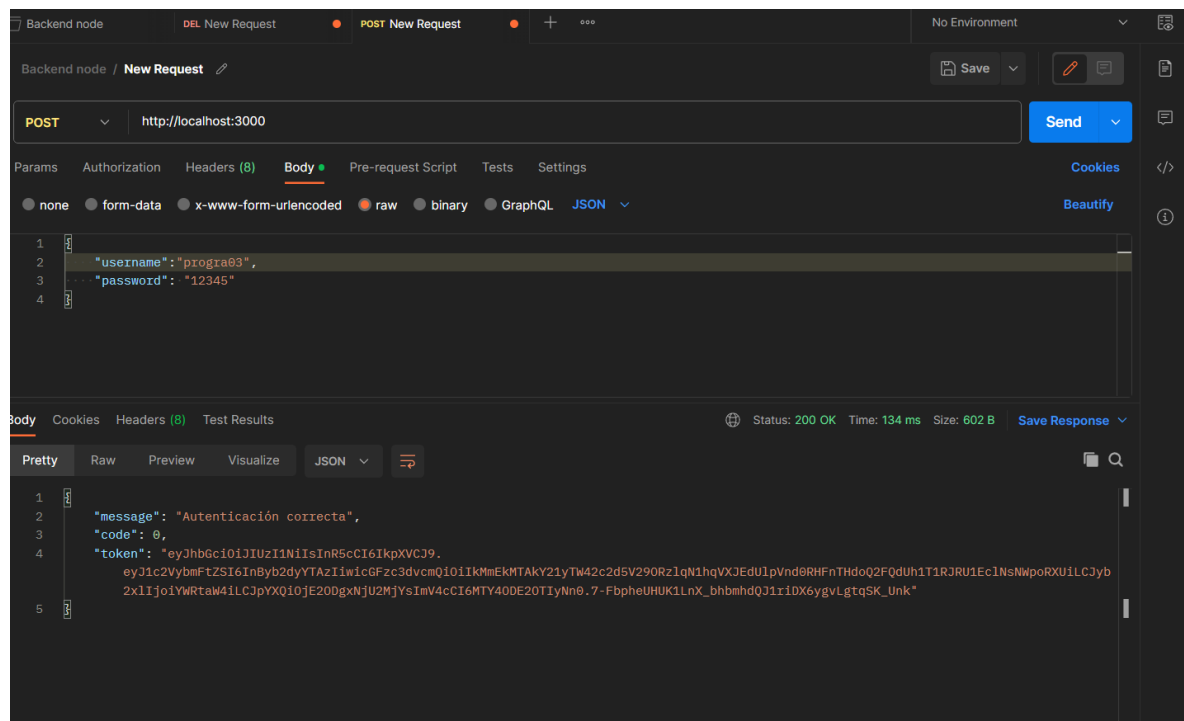
Se importa jwt.

```
import model from '../models/authModelo'
import jwt from 'jsonwebtoken';
import { utils } from "../utils/utils";
import conn from '../config/dbmysql'
```

Termine en clase esta parte.

```
if(await utils.checkPassword(password, lstUsers[0].password)){
  const newUser = {
    username: lstUsers[0].username,
    password: lstUsers[0].password,
    role: lstUsers[0].role
  }
  console.log(newUser)
  let token= jwt.sign(newUser, conn.keys.secret, {expiresIn:'1h'});

  return res.json({message:"Autenticación correcta",
    code:0,
    token:token
  })
}
```



Decodificando token.

The screenshot shows the JWT.io website interface. The top navigation bar includes links for 'Debugger', 'Libraries', 'Introduction', and 'Ask', along with a 'Crafted by auth0' logo. The main content area is split into two panels: 'Encoded' on the left and 'Decoded' on the right.

Encoded Panel: A text input field contains the following JWT token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InByb2dyYTZlcnR5c3dvc2d5V290Rz1qN1hqVXJlZUlpVnd0RHFnTHdoQ2FqdUhh1T1RJRU1Ec1N5NWp0RXU1LCJyb2x1Ijo1YWRTaW41LCJpYXQ1OjE2ODgxNjU2MjYsImV4cCI6MTY4ODE2OTIyNm0.7FbpheUHUK1LnX_bhbmhdQJ1riDX6ygvLgtqSK_Unk`. Below the input is an 'Explicar' button.

Decoded Panel: This panel displays the decoded structure of the token.

- HEADER: ALGORITHM & TOKEN TYPE:**

```
{  "alg": "HS256",  "typ": "JWT"}
```
- PAYLOAD: DATA:**

```
{  "username": "progra83",  "password": "52a510$cmrMn6sgyW0N09j7XJUrDuIiVvtDqgLwhCaPuhU0TIEhDrS15jhEu",  "role": "admin",  "iat": 1688165626,  "exp": 1688169226}
```
- VERIFY SIGNATURE:** Shows the HMACSHA256 formula: `HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), your-256-bit-secret)`. Below the formula is a checkbox labeled 'secret base64 encoded'.

The bottom of the image shows a Windows taskbar with the date and time '04:57 p.m. 30/06/2023'.