

Instalar [TypeScript](#)

```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Versión 10.0.22621.1848]
(c) Microsoft Corporation. Todos los derechos reservados.

D:\Proyectos\Unidad>npm install -g typescript

changed 1 package in 18s

D:\Proyectos\Unidad>tsc -v
Version 5.1.6

D:\Proyectos\Unidad>
```

Checar la versión de TypeScript

Ubicarse en una carpeta de trabajo para la creación del BackeEnd con NodeJs y Express

Crear una carpeta backend_mysql

Cambiarse a carpeta backend_mysql recién creada,

```
C:\WINDOWS\system32\cmd. X + v
D:\Proyectos\Unidad>npm install -g typescript

changed 1 package in 18s

D:\Proyectos\Unidad>tsc -v
Version 5.1.6

D:\Proyectos\Unidad>mkdir backend_mysql

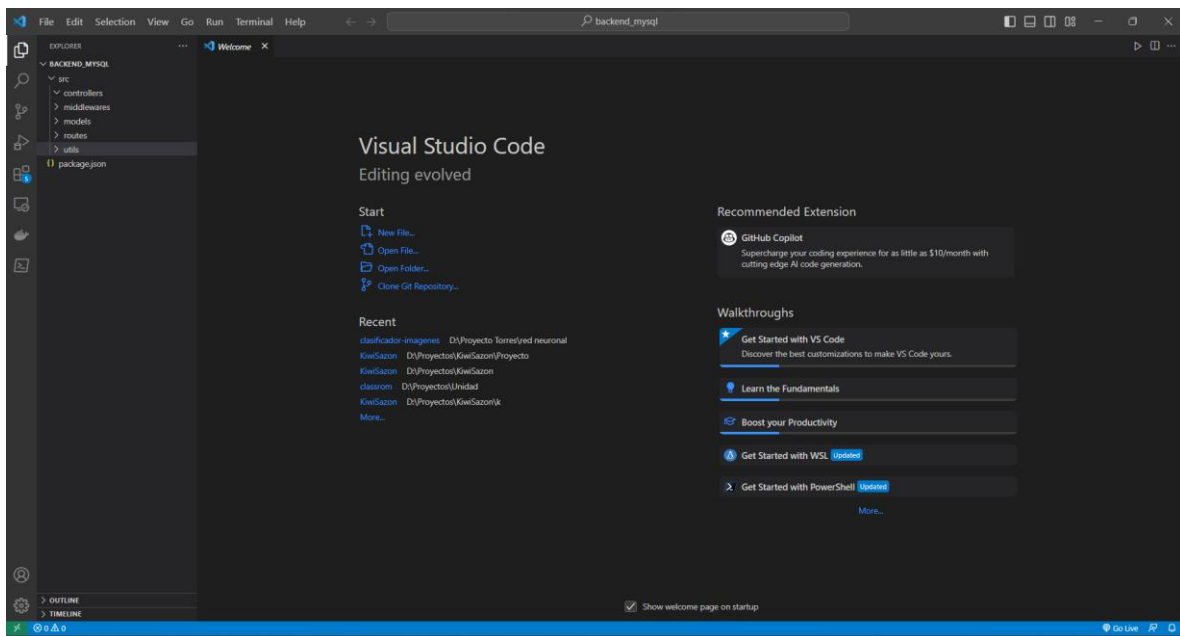
D:\Proyectos\Unidad>cd backend_mysql

D:\Proyectos\Unidad\backend_mysql>npm init -y
Wrote to D:\Proyectos\Unidad\backend_mysql\package.json:

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

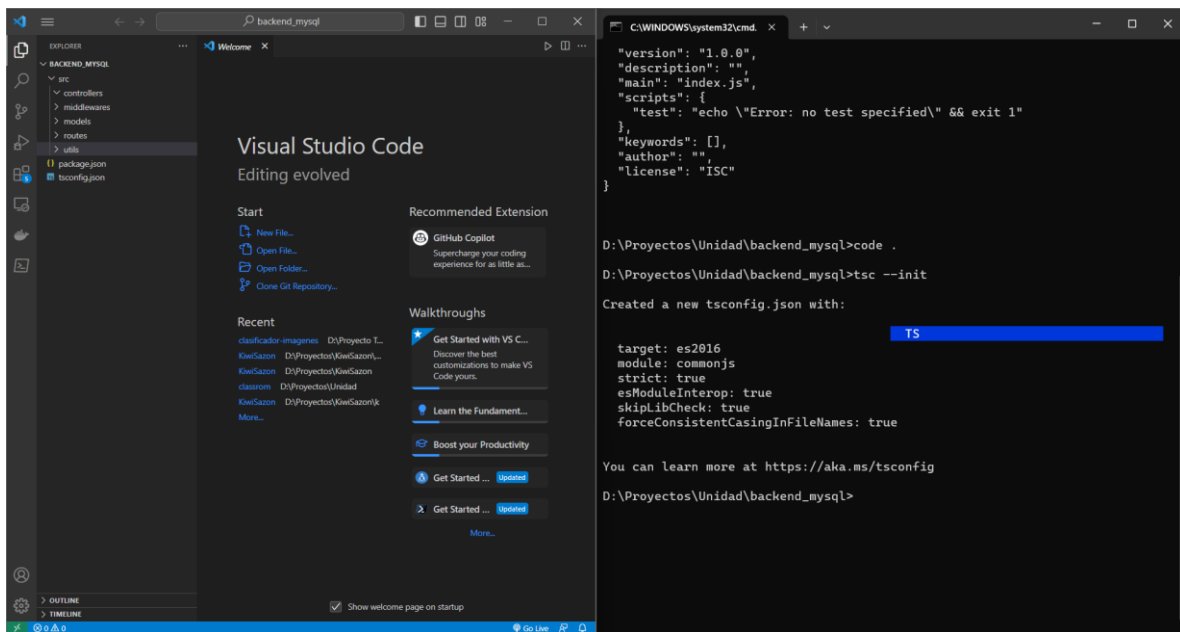
D:\Proyectos\Unidad\backend_mysql>
```

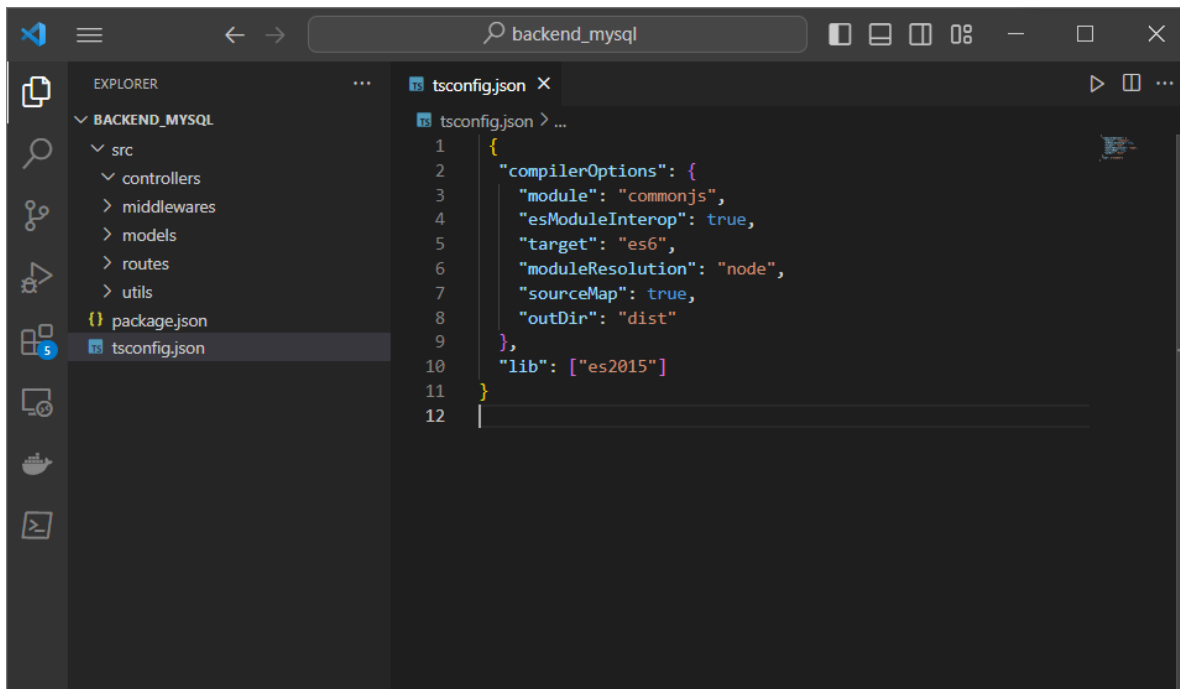
Abrir carpeta en VisualStudio Code



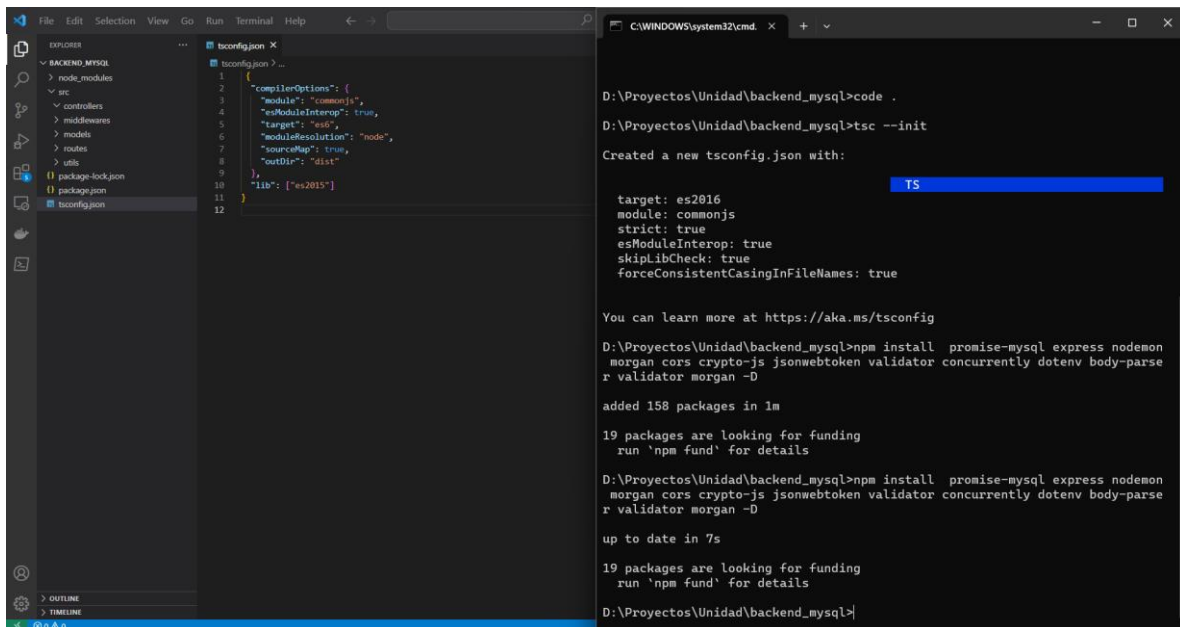
Dentro de la carpeta ejecutar el siguiente comando para creación de archivo package.json

`npm init -y`

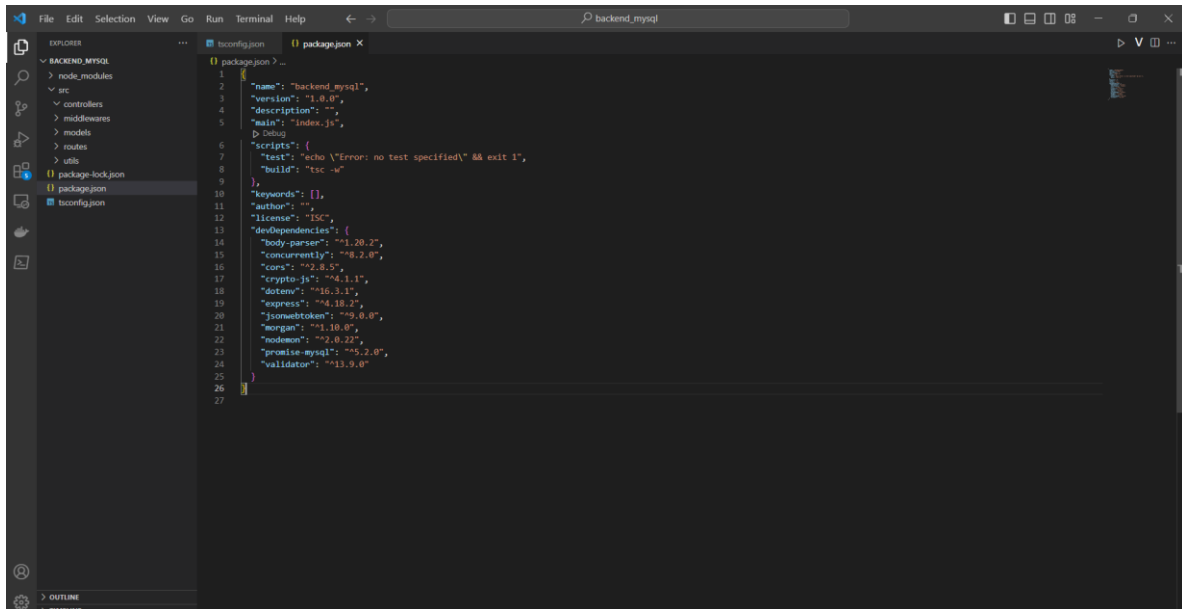




Instalar dependencias



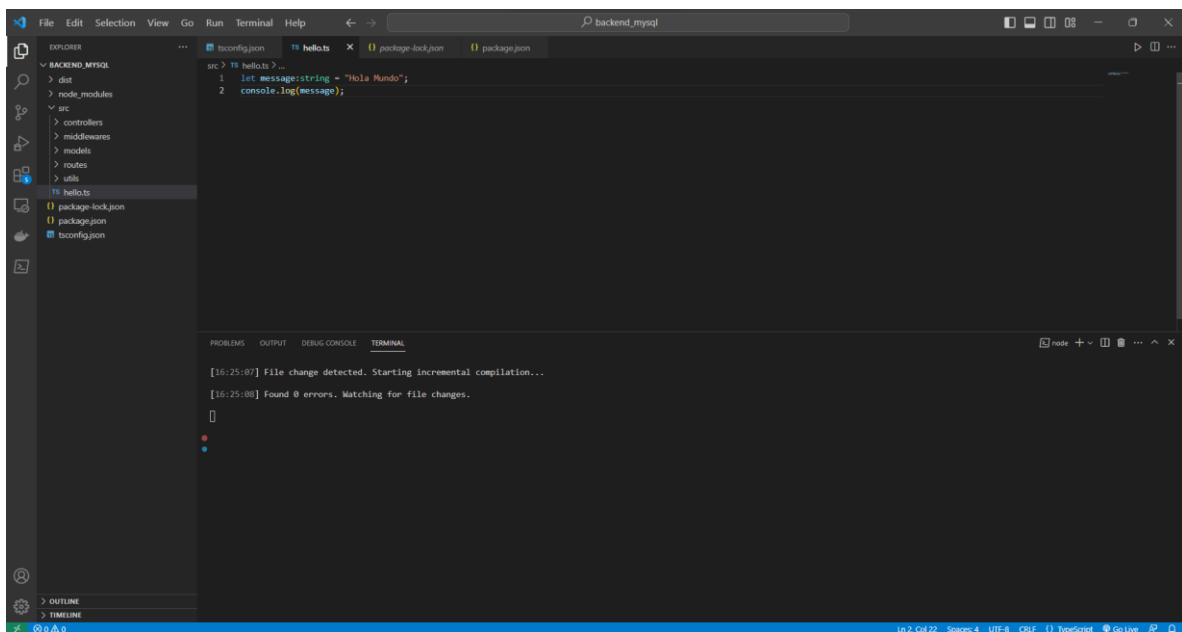
Modificar el archivo package.json para agregar el script build



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

```
1 {
2   "name": "backend_mysql",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1",
8     "build": "tsc -b"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "devDependencies": {
14    "body-parser": "^1.20.2",
15    "concurrently": "^8.2.0",
16    "cors": "^2.8.5",
17    "crypto-js": "^4.1.1",
18    "dotenv": "^16.3.1",
19    "express": "^4.18.2",
20    "jsonwebtoken": "^9.0.0",
21    "morgan": "^1.10.0",
22    "nodemon": "^2.0.22",
23    "prisma-mysql": "^1.2.0",
24    "validator": "^13.9.0"
25  }
26 }
```

Crear un archivo src/hello.ts



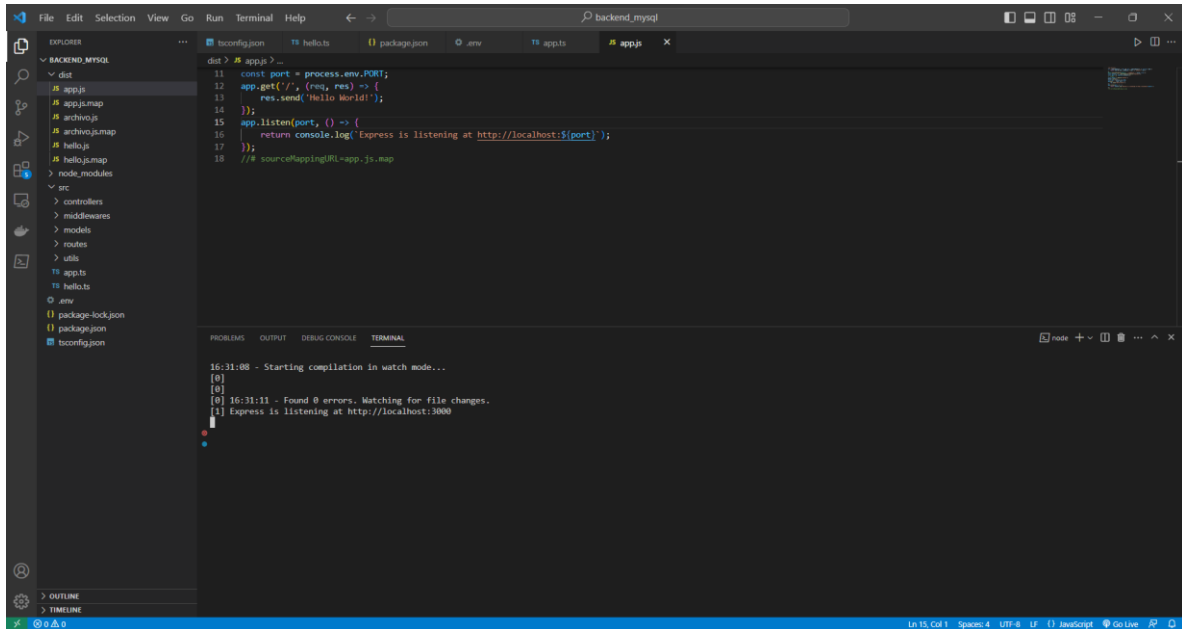
The screenshot shows the Visual Studio Code editor with the `src/hello.ts` file open. The file contains the following TypeScript code:

```
1 let message:string = "Hola Mundo";
2 console.log(message);
```

The terminal output shows the following messages:

```
[16:25:07] File change detected. Starting incremental compilation...
[16:25:08] Found 0 errors. Matching for file changes.
[]
```

Crear archivo src/app.ts, es nuestro punto de inicio de nuestra aplicación



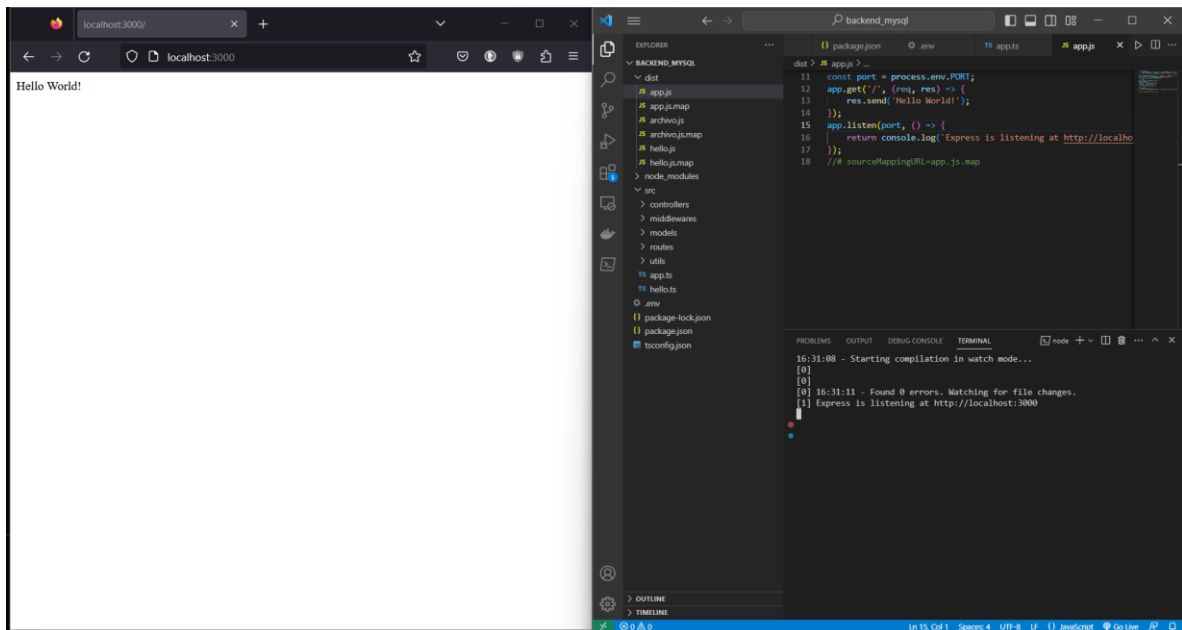
The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the project structure. The file `src/app.ts` is selected. The editor window shows the following code:

```
dist > appjs > ...
11 const port = process.env.PORT;
12 app.get('/', (req, res) => {
13   res.send('Hello World!');
14 });
15 app.listen(port, () => {
16   return console.log('Express is listening at http://localhost:5(port)');
17 });
18 /** sourceMappingURL=app.js.map
```

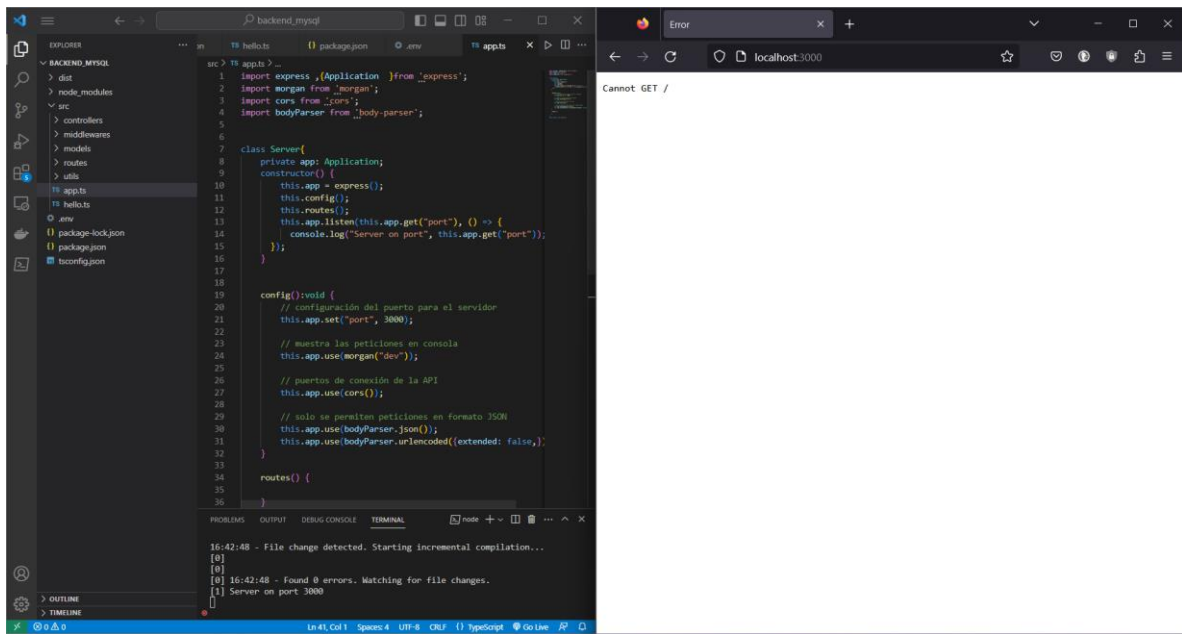
The terminal at the bottom shows the following output:

```
16:31:08 - Starting compilation in watch mode...
[0]
[0] 16:31:11 - Found 0 errors. Watching for file changes.
[1] Express is listening at http://localhost:3000
```

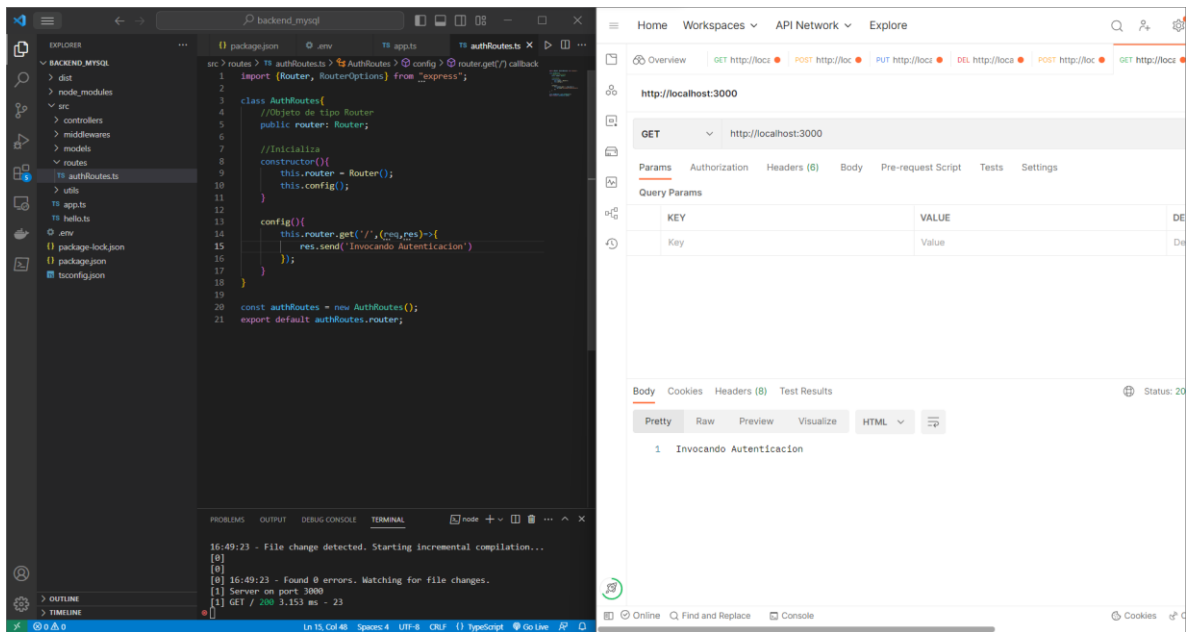
Verificando el funcionamiento



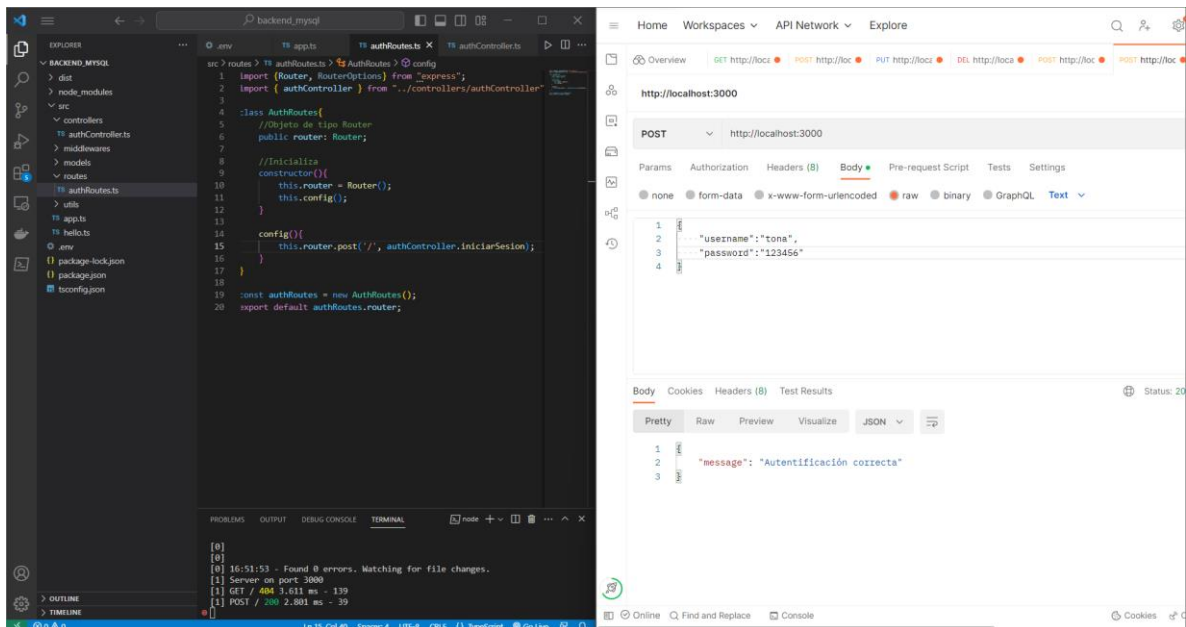
The screenshot shows a web browser window at `localhost:3000/` displaying the text "Hello World!". The background shows the Visual Studio Code editor with the same code as in the previous screenshot. The terminal output is the same as well.



Verificar path



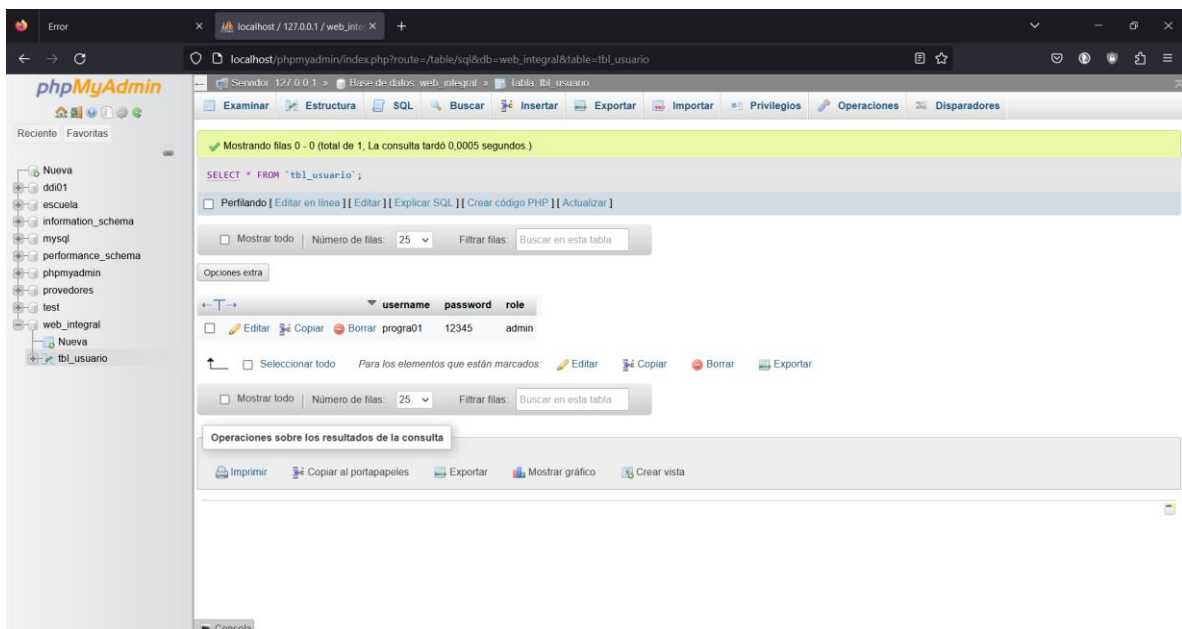
Verificar envío de datos nuevamente con Postman



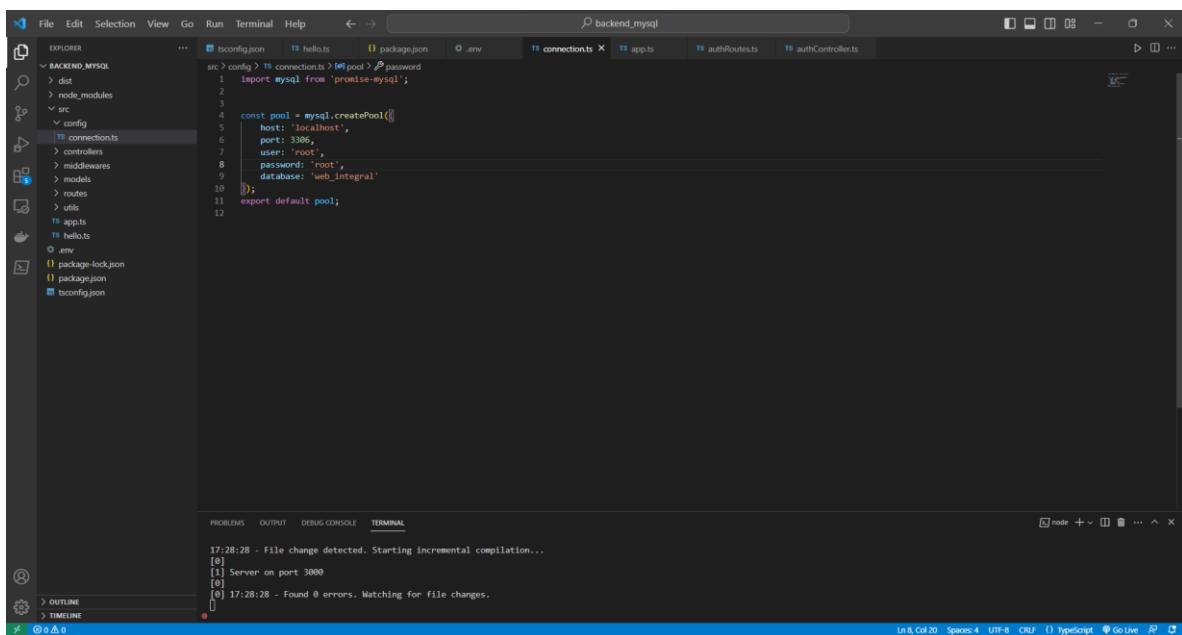
Crear una base de datos en MySQL llamada web_integral

Crear una tabla llamada tbl_usuario

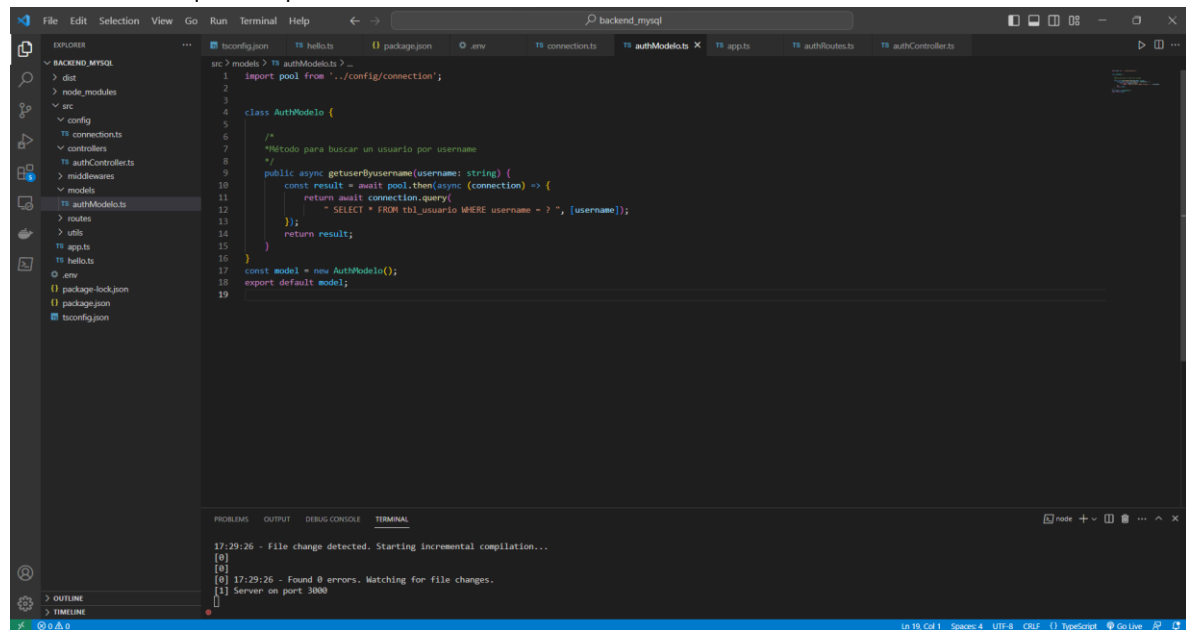
Insertar el usuario de acuerdo a los siguientes datos



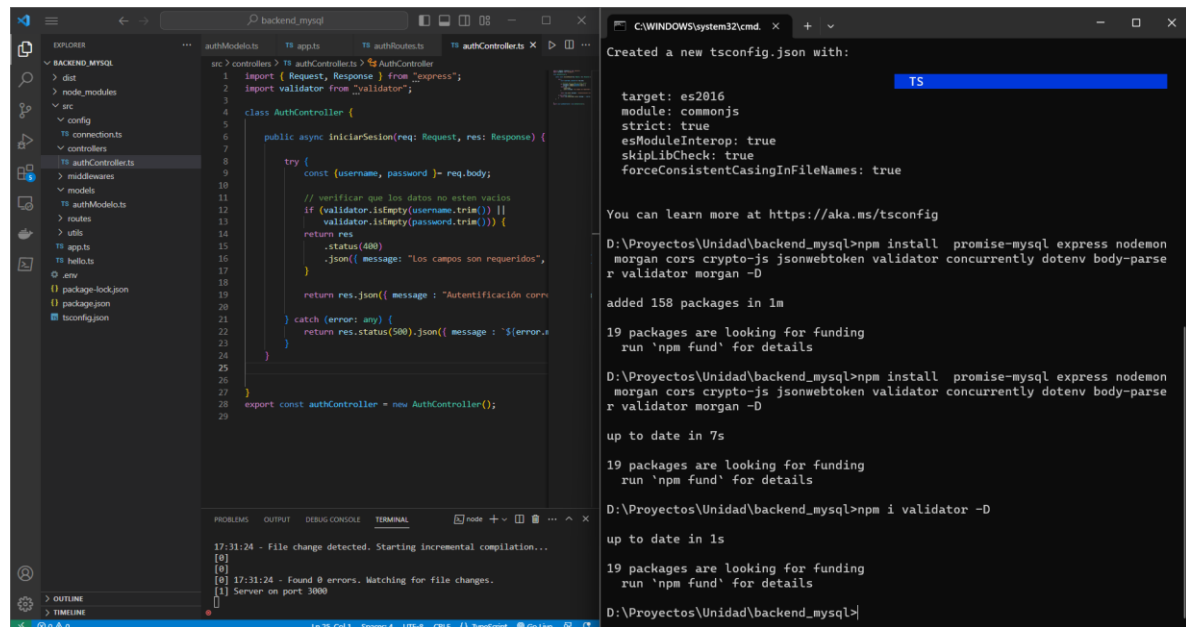
Crear archivo para conexión a base de datos MySQL [src|config|connection.ts](#)



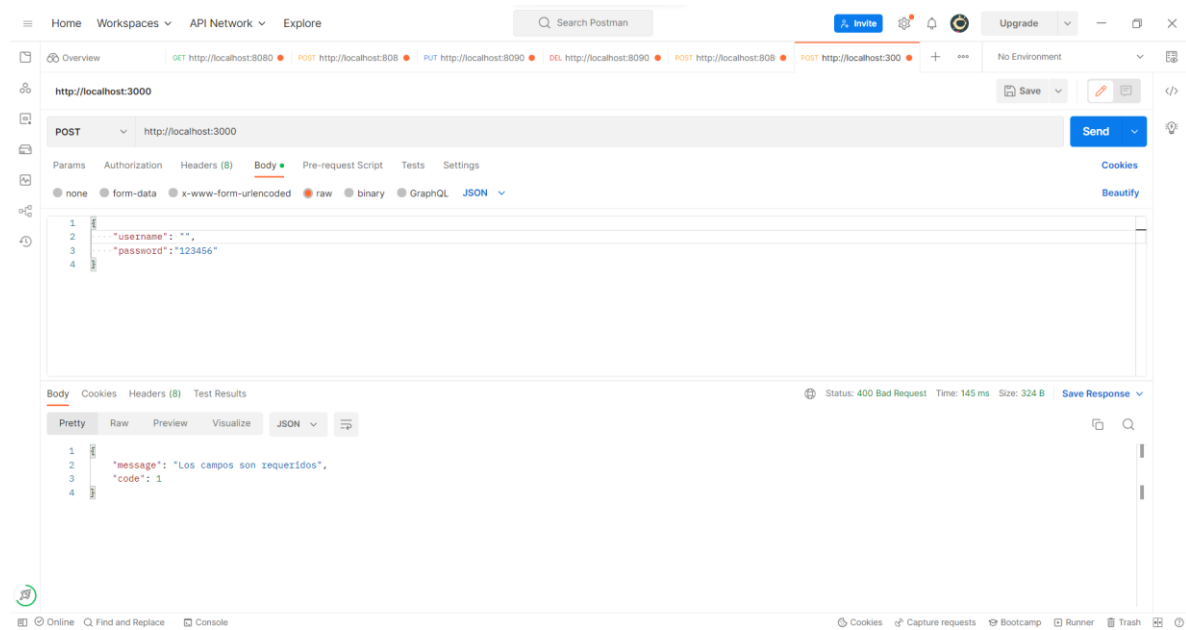
Crear archivo src\models\authModelo.ts



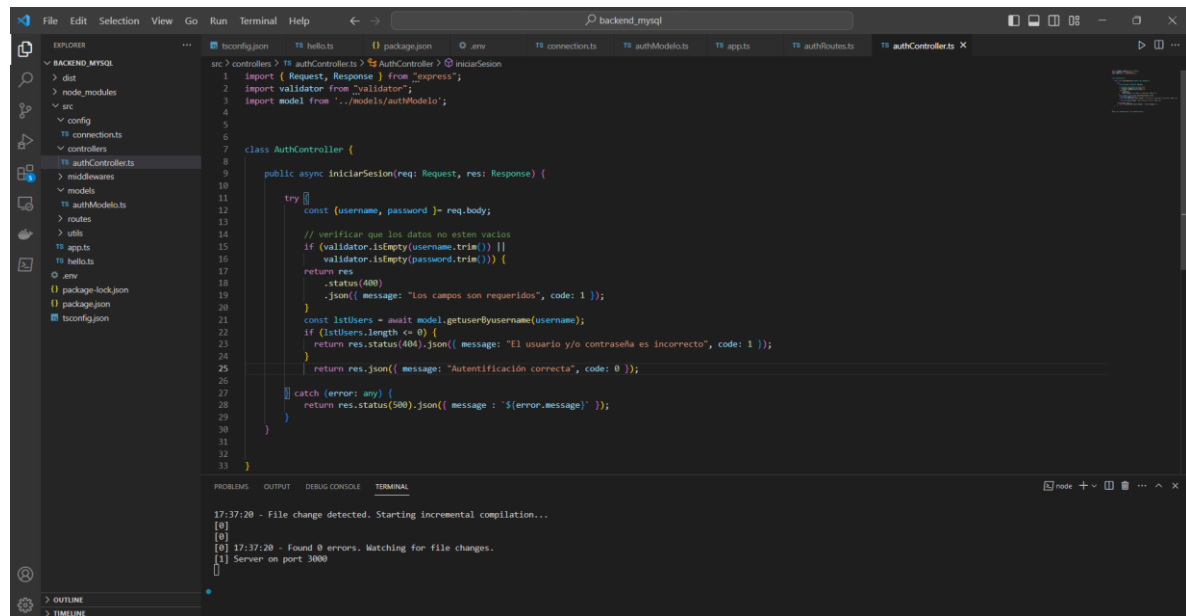
Modificar archivo src\controllers\authController



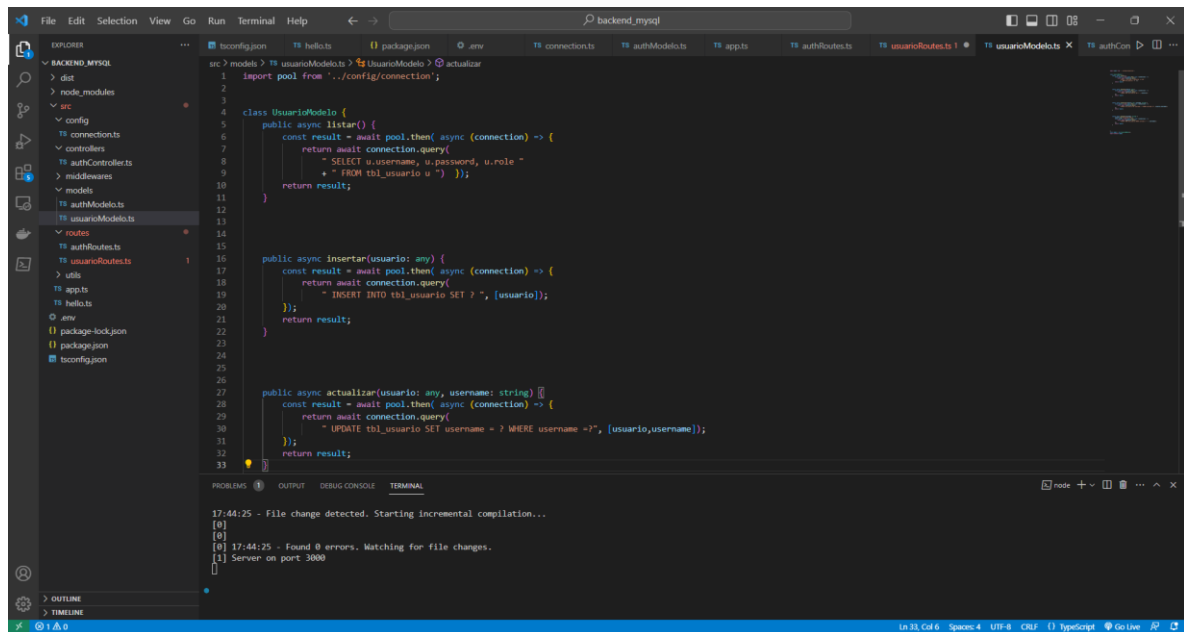
Verificar con Postman



Definiendo el acceso a los datos de datos por lo qué es necesario



Crear el modelo usuarioModelo.ts en carpeta models

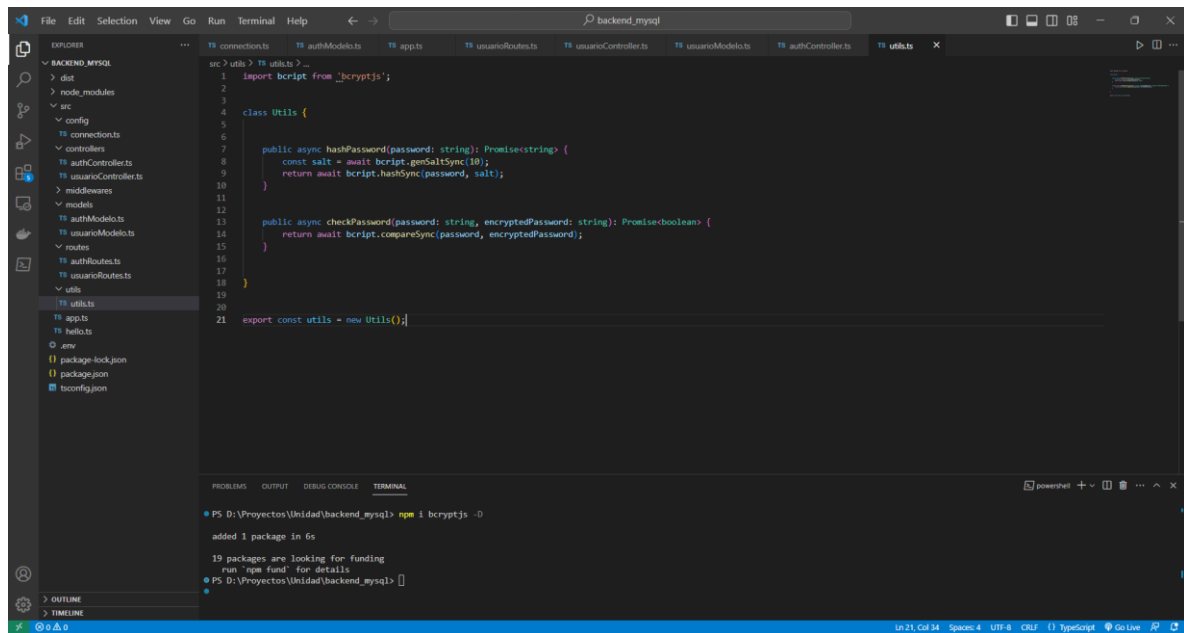


The screenshot shows the Visual Studio Code editor with the 'usuarioModelo.ts' file open in the 'models' folder. The code defines a class 'UsuarioModelo' with three methods: 'listar()', 'insertar(usuario: any)', and 'actualizar(usuario: any, username: string)'. The 'listar()' method uses a database pool to fetch users from 'tbl_usuario'. The 'insertar()' method inserts a new user into 'tbl_usuario'. The 'actualizar()' method updates a user's username in 'tbl_usuario'. The terminal shows the server starting on port 3000.

```
src > models > TS usuarioModelo.ts > actualizar
1 import pool from '../config/connection';
2
3
4 class UsuarioModelo {
5     public async listar() {
6         const result = await pool.then(async (connection) => {
7             return await connection.query(
8                 "SELECT username, u.password, u.role "
9                 + "FROM tbl_usuario u " );
10            return result;
11        });
12    }
13
14    public async insertar(usuario: any) {
15        const result = await pool.then(async (connection) => {
16            return await connection.query(
17                "INSERT INTO tbl_usuario SET ? ", [usuario]);
18            });
19        return result;
20    }
21
22    public async actualizar(usuario: any, username: string) {
23        const result = await pool.then(async (connection) => {
24            return await connection.query(
25                "UPDATE tbl_usuario SET username = ? WHERE username = ?", [usuario, username]);
26            });
27        return result;
28    }
29
30 }
31
32 export const usuarioModelo = new UsuarioModelo();
33
```

17:44:25 - File change detected. Starting incremental compilation...
[0]
[0]
17:44:25 - Found 0 errors. Matching for file changes.
[1] Server on port 3000

Crear archivo src|utils|utils.ts

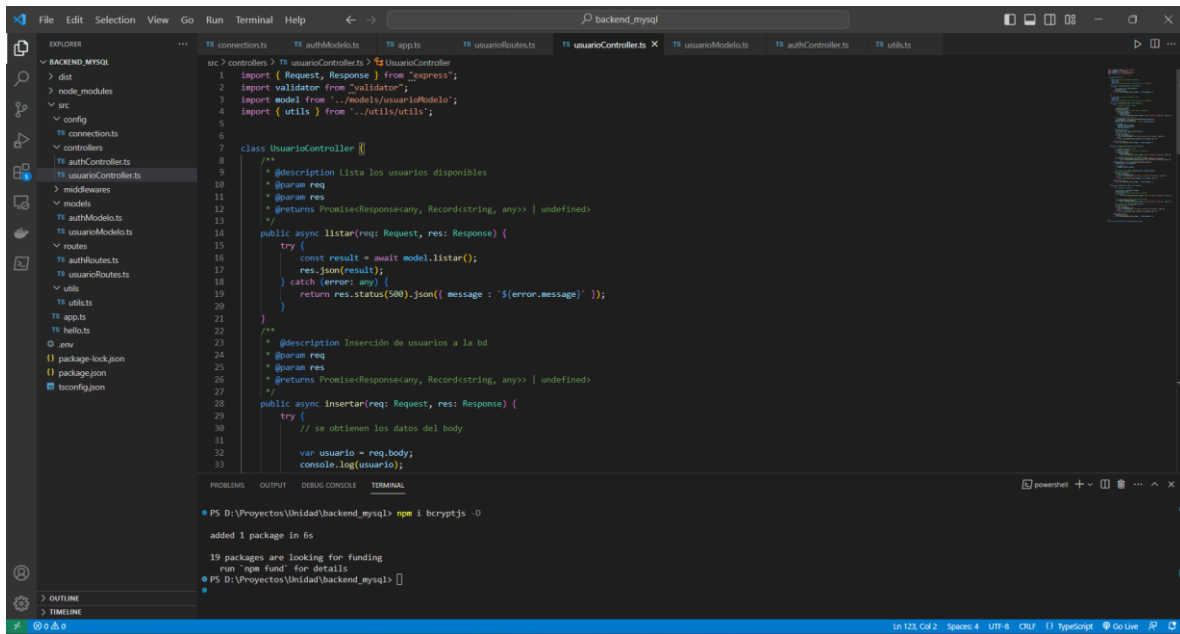


The screenshot shows the Visual Studio Code editor with the 'utils.ts' file open in the 'src|utils' folder. The code defines a class 'Utils' with two methods: 'hashPassword(password: string): Promise<string>' and 'checkPassword(password: string, encryptedPassword: string): Promise<boolean>'. The 'hashPassword()' method uses 'bcrypt' to hash a password. The 'checkPassword()' method uses 'bcrypt' to compare a password with its encrypted version. The terminal shows the installation of 'bcryptjs' and the server starting on port 3000.

```
src > utils > TS utils.ts >
1 import bcrypt from 'bcryptjs';
2
3
4 class Utils {
5
6
7     public async hashPassword(password: string): Promise<string> {
8         const salt = await bcrypt.genSaltSync(10);
9         return await bcrypt.hashSync(password, salt);
10    }
11
12    public async checkPassword(password: string, encryptedPassword: string): Promise<boolean> {
13        return await bcrypt.compareSync(password, encryptedPassword);
14    }
15
16 }
17
18 export const utils = new Utils();
19
20
21 export const utils = new Utils();
22
```

PS D:\Proyectos\Unidad\backend_mysql> npm i bcryptjs -D
added 1 package in 6s
19 packages are looking for funding
run 'npm fund' for details
PS D:\Proyectos\Unidad\backend_mysql>

Controlador usuarioController



```
src > controllers > TS usuarioControllers > TS UsuarioController
1 import { Request, Response } from "express";
2 import validator from "validator";
3 import model from "../models/usuarioModelo";
4 import { utils } from "../utils/utils";
5
6
7 class UsuarioController {
8
9   /**
10    * @description Lista los usuarios disponibles
11    * @param req
12    * @param res
13    * @returns Promise<Responseany, Record<string, any>> | undefined
14    */
15   public async listar(req: Request, res: Response) {
16     try {
17       const result = await model.listar();
18       res.json(result);
19     } catch (error: any) {
20       return res.status(500).json({ message: `${error.message}` });
21     }
22   }
23
24   /**
25    * @description Inserción de usuarios a la bd
26    * @param req
27    * @param res
28    * @returns Promise<Responseany, Record<string, any>> | undefined
29    */
30   public async insertar(req: Request, res: Response) {
31     try {
32       // se obtienen los datos del body
33       var usuario = req.body;
34       console.log(usuario);
35     }
36   }
37 }
```

PS D:\Proyectos\Unidad\backend_mysql> npm i bcryptjs -D

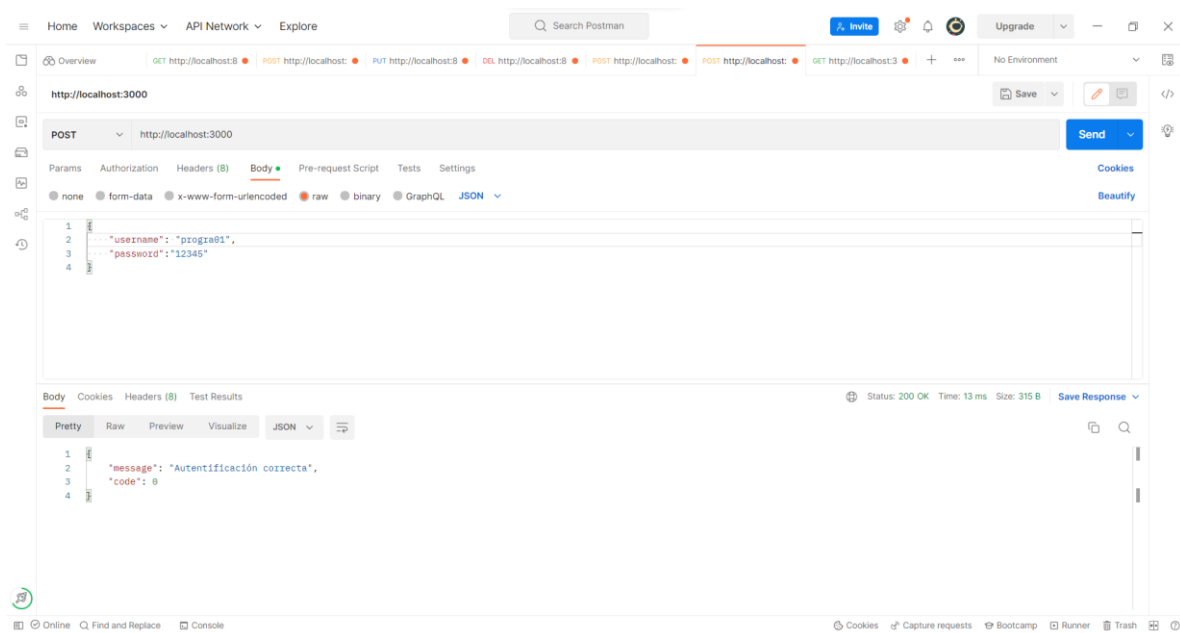
added 1 package in 6s

19 packages are looking for funding

run 'npm fund' for details

PS D:\Proyectos\Unidad\backend_mysql>

Realizar las pruebas con Postman



Home Workspaces API Network Explore Search Postman

Overview GET http://localhost:8 POST http://localhost: PUT http://localhost: DEL http://localhost: POST http://localhost: POST http://localhost: GET http://localhost:3 No Environment

http://localhost:3000/user/ Save Send

GET http://localhost:3000/user/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	*** Bulk Edit
Key	Value	Description	

Body Cookies Headers (8) Test Results Status: 200 OK Time: 35 ms Size: 326 B Save Response

Pretty Raw Preview Visualize JSON Bulk Edit

```
1 {
2   "username": "progra01",
3   "password": "12345",
4   "role": "admin"
5 }
```

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

Home Workspaces API Network Explore Search Postman

Overview GET http://localhost: POST http://localhost: PUT http://localhost: DEL http://localhost: POST http://localhost: POST http://localhost: GET http://localhost: POST http://localhost: No Environment

http://localhost:3000/user/ Save Send

POST http://localhost:3000/user/ 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": "tona",
3   "password": "123456",
4   "role": "admin"
5 }
```

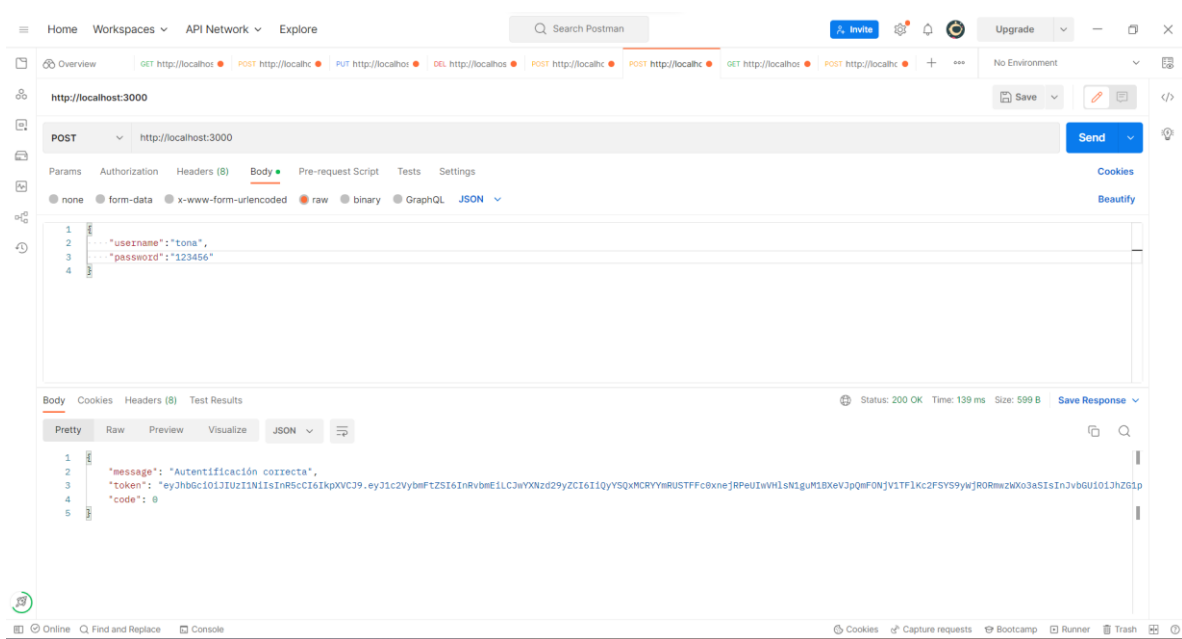
Body Cookies Headers (8) Test Results Status: 200 OK Time: 104 ms Size: 326 B Save Response

Pretty Raw Preview Visualize JSON Bulk Edit

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

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

Una vez que se encuentra el usuario y la contraseña verifica con el módulo **bcrypt** si coincide la contraseña



Error

localhost / 127.0.0.1 / web_int...

JSON Web Tokens - jwt.io

+

←

→

↻

https://jwt.io

☆


🔍

🔔


🔧

📁

☰

 JWT

Debugger Libraries Introduction Ask

Crafted by  auth0 by OKTA

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InRvbmEiLCJwYXNzd29yZCI6IiQyYSQxMCRYYmRUSTFFc0xnejRPeUIwVHlsN1guM1BxeVJpQmFONjV1TF1Kc2FSYS9yWjRORmwzWXo3aSI6InJvbGUiOiJhZG1pb2I6Im1hdCI6MTY4ODU0MDA1NiwiZXhwIjoxNjg4NTQzNjU2fQ.H-cbqD-b4XViiX9o5y4JqCzQhV5dLzQbwPfmKu2XV9E
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "username": "tona",
  "password":
    "S2a$10$XbdIT1EsLgz40yB8Ty17X.3PWYr1BaN65uLYJsaRa/rZ4NF13Yz7i",
  "role": "admin",
  "iat": 1688540056,
  "exp": 1688543656
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + ".",
  base64UrlEncode(payload),
  your-256-bit-secret
)
```

☐ secret base64 encoded