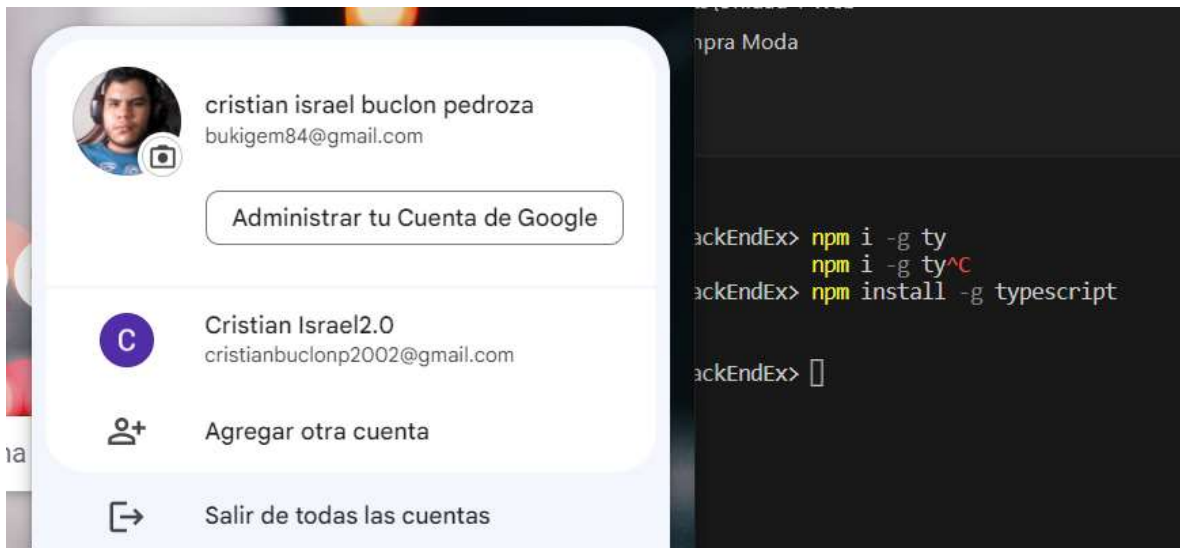


## Back End Cristian Israel Buclon Pedroza

### 1. Instalación del Typescript.




### 2. Versión TSC



### 3. Creacion de carpeta Backend\_mysql



### 4. Cambio de carpeta backend\_mySQL



cristian israel buclon pedroza  
bukigem84@gmail.com

Administrar tu Cuenta de Google

Cristian Israel2.0  
cristianbuclonp2002@gmail.com

Agregar otra cuenta


```
unidad 3 BackEnd\BackEndEx> tsc -v
unidad 3 BackEnd\BackEndEx> mkdir backend_mysql

LL\Documents\unidad 3 BackEnd\BackEndEx

iteTime      Length Name
-----
6 p. m.      backend_mysql

unidad 3 BackEnd\BackEndEx> cd backend_mysql
unidad 3 BackEnd\BackEndEx\backend_mysql> 
```

##### 5. Ejecución del comando npm init -y



cristian israel buclon pedroza  
bukigem84@gmail.com

Administrar tu Cuenta de Google

Cristian Israel2.0  
cristianbuclonp2002@gmail.com

```
Length Name
-----
backend_mysql

BackEnd\BackEndEx> cd backend_mysql
BackEnd\BackEndEx\backend_mysql> npm init -y
```

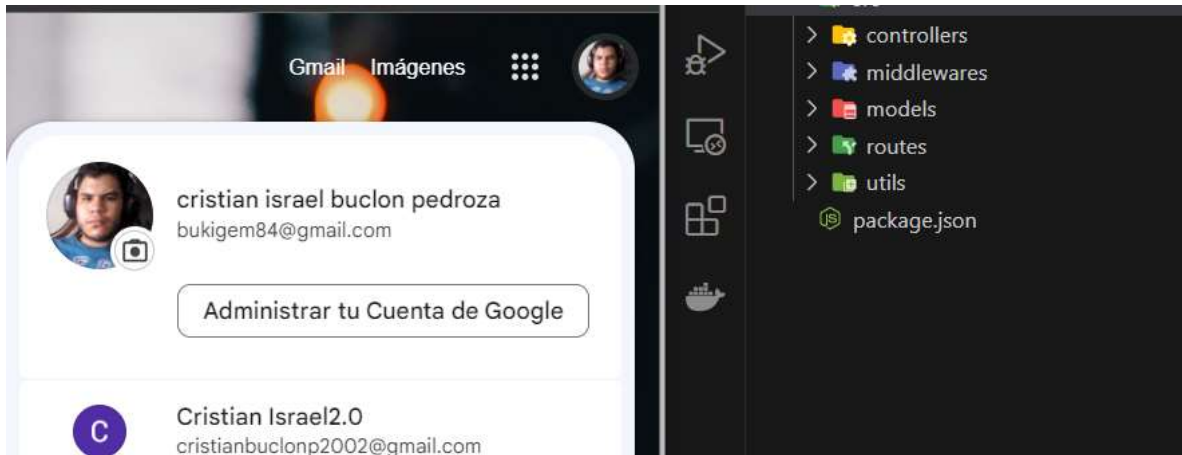
Welcome

BACKENDX

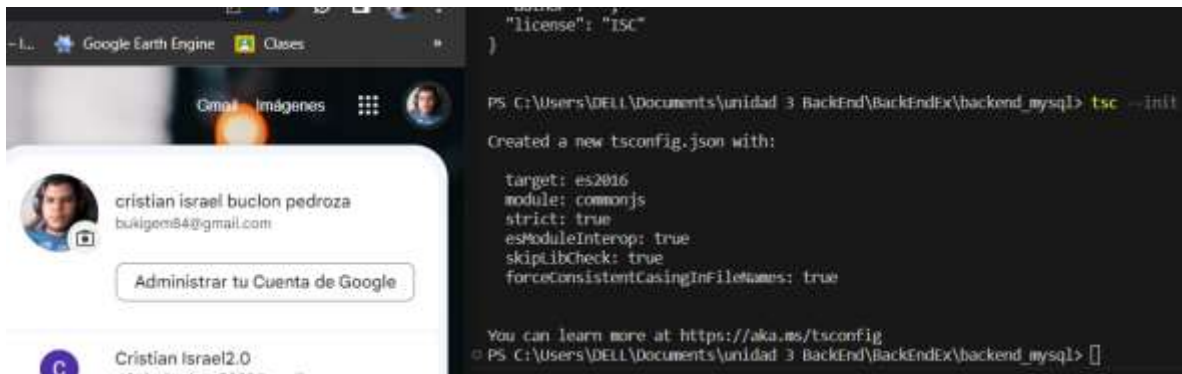
backend\_mysql

package.json

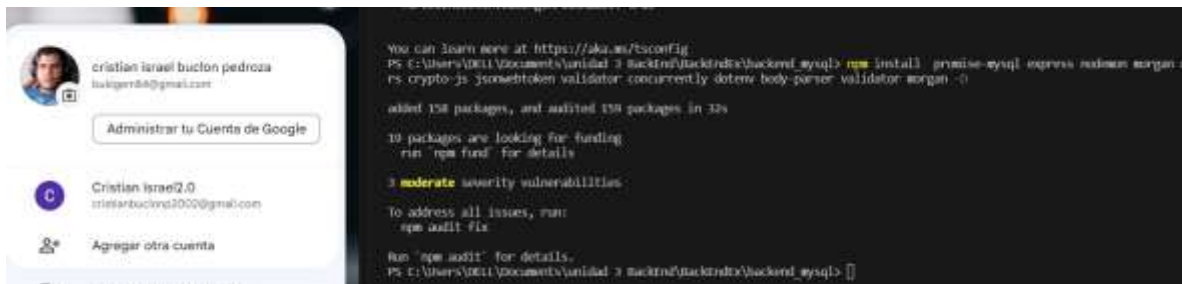
##### 6. Estructura de carpetas



## 7. Archivo configuración ts



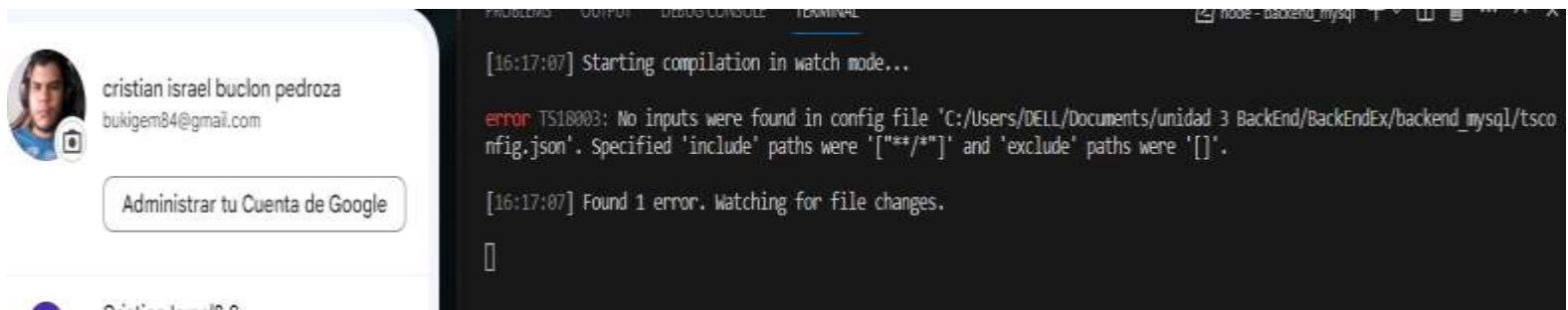
## 8. Instalacion de Express y de Node



## 9. Buid TSC



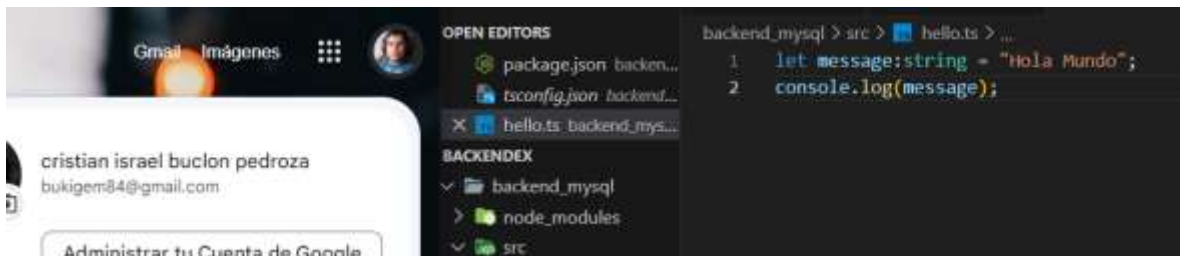
## 10. Ejecución de npm run build



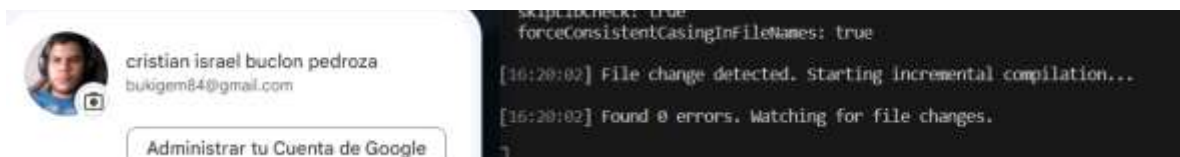
#### 11. Reemplazo del archivo config



#### 12. Creacion de Archivo Hello.js



#### 13. Comprobación de errores inexistentes



#### 14. Agregado de script Start



#### 15. Creación .env



cristian israel buclon pedroza  
bukigem84@gmail.com

```
.env  
1 PORT = 3000
```

## 16. Creación app.ts

The screenshot shows a web browser interface on the left with Google account management options for 'cristian israel buclon pedroza' and 'Cristian Israel2.0'. On the right, a code editor displays the contents of 'app.ts' and 'package.json'.

```
backend_mysql > src > app.ts > ...  
1 import express from 'express';  
2 const app = express();  
3  
4 /* DotEnv */  
5 const dotenv = require('dotenv');  
6 dotenv.config();  
7 const port = process.env.PORT;  
8  
9 app.get('/', (req, res) => {  
10   res.send('Hello World!');  
11 });  
12  
13 app.listen(port, () => {  
14   return console.log('Express is listening at http://localhost:${port}');  
15 });  
16
```

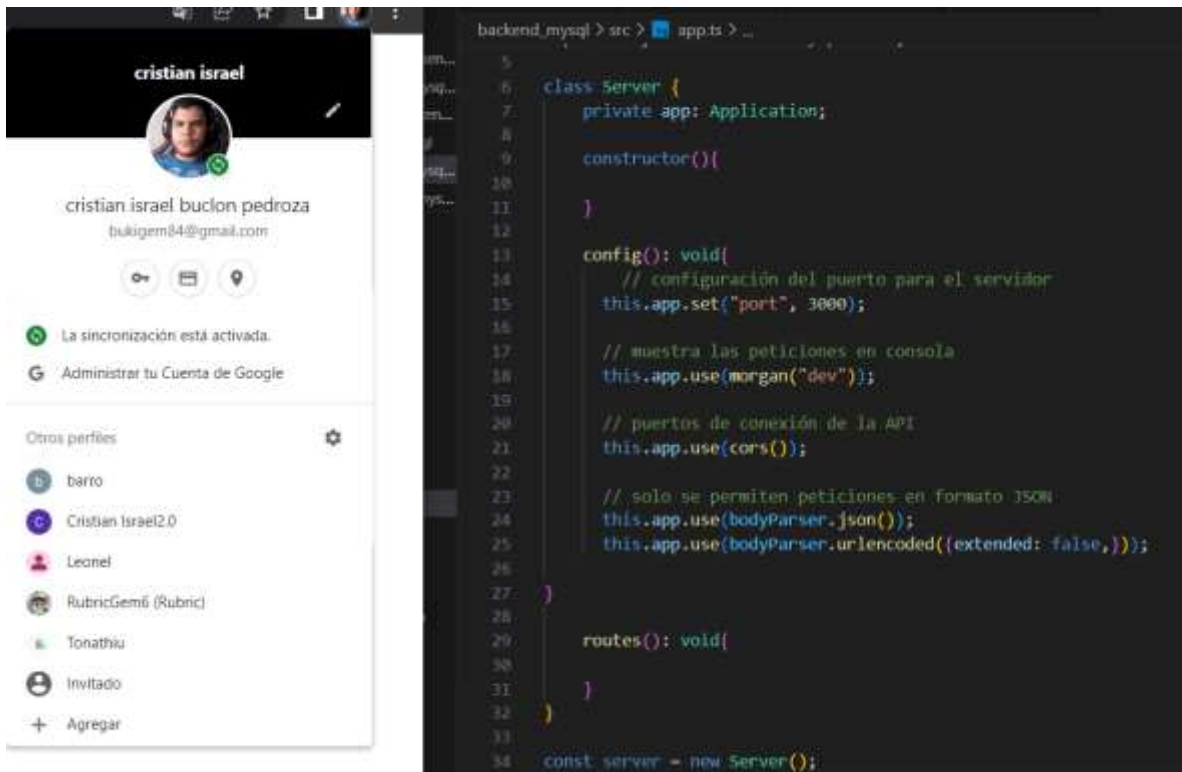
```
{  
  "name": "app",  
  "version": "1.0.0",  
  "description": "",  
  "main": "dist/app.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1",  
    "start": "node build/app.js",  
    "dev": "concurrently \\\"npx tsc --watch\\\" \\\"nodemon -q dist/app.js\\\"",  
  },  
  "keywords": [],  
  "author": ""  
}
```

## 17: Comprobación de mensaje Hello World

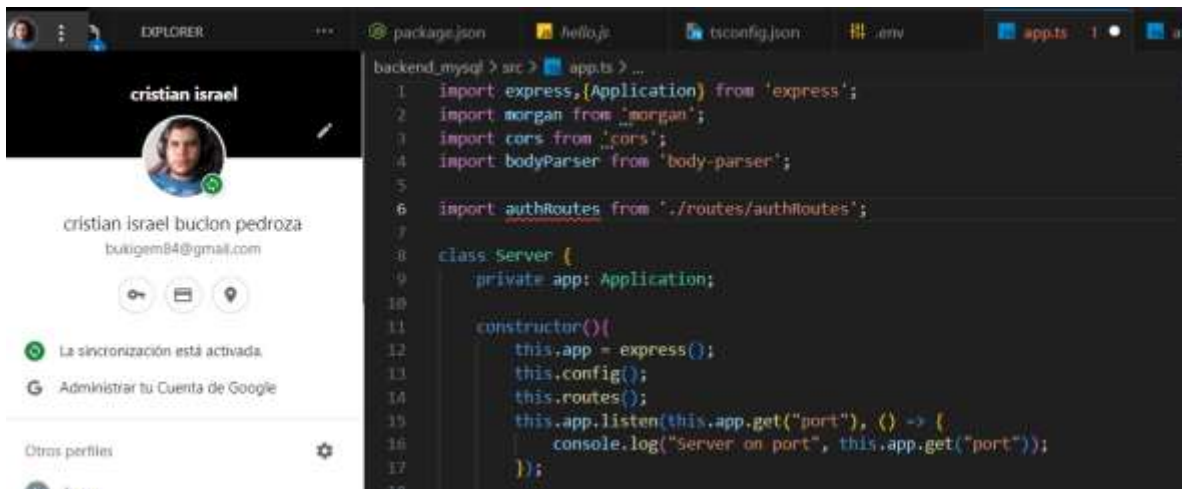
This screenshot displays the 'app.ts' code on the left, the 'Hello World!' message rendered in the browser's main content area in the center, and the Google account management interface on the right, which includes a profile picture and the name 'cristian israel'.

## 18. Definición de clase server en el app.ts

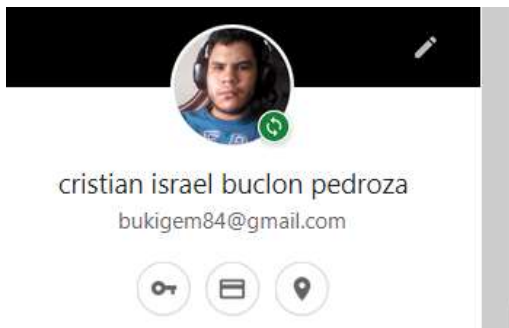




## 19. Constructor de del puerto



## 20. Auth Routes



```

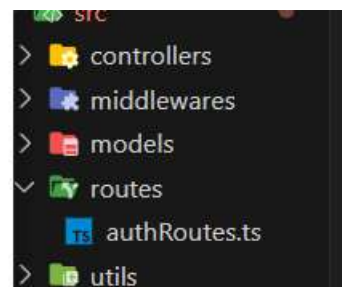
}

routes(): void{
  this.app.use("/", authRoutes);
}

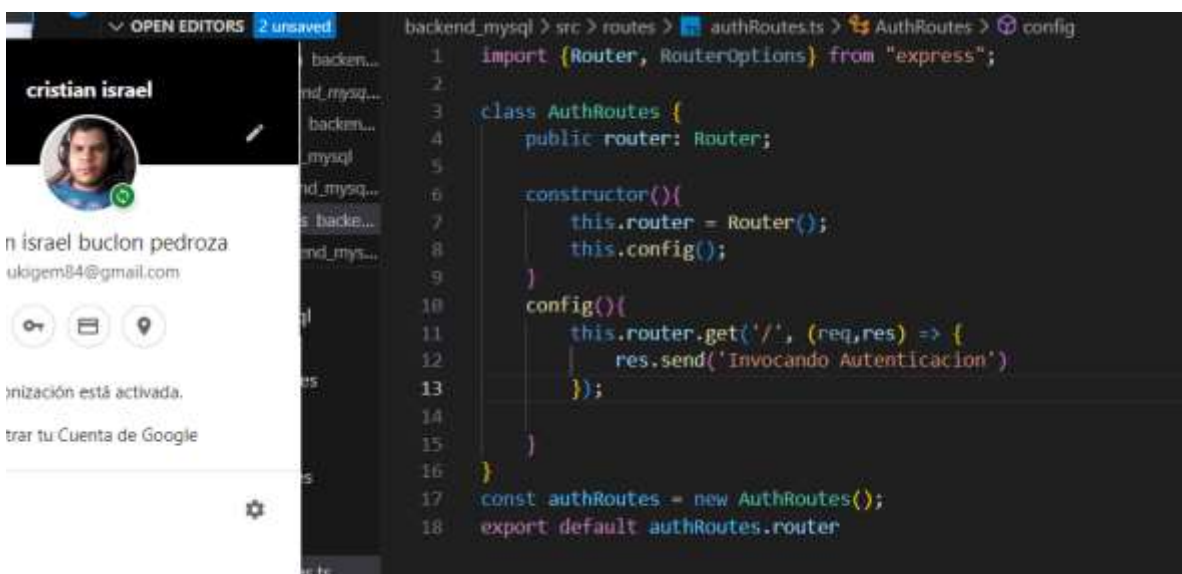
const server = new Server();

```

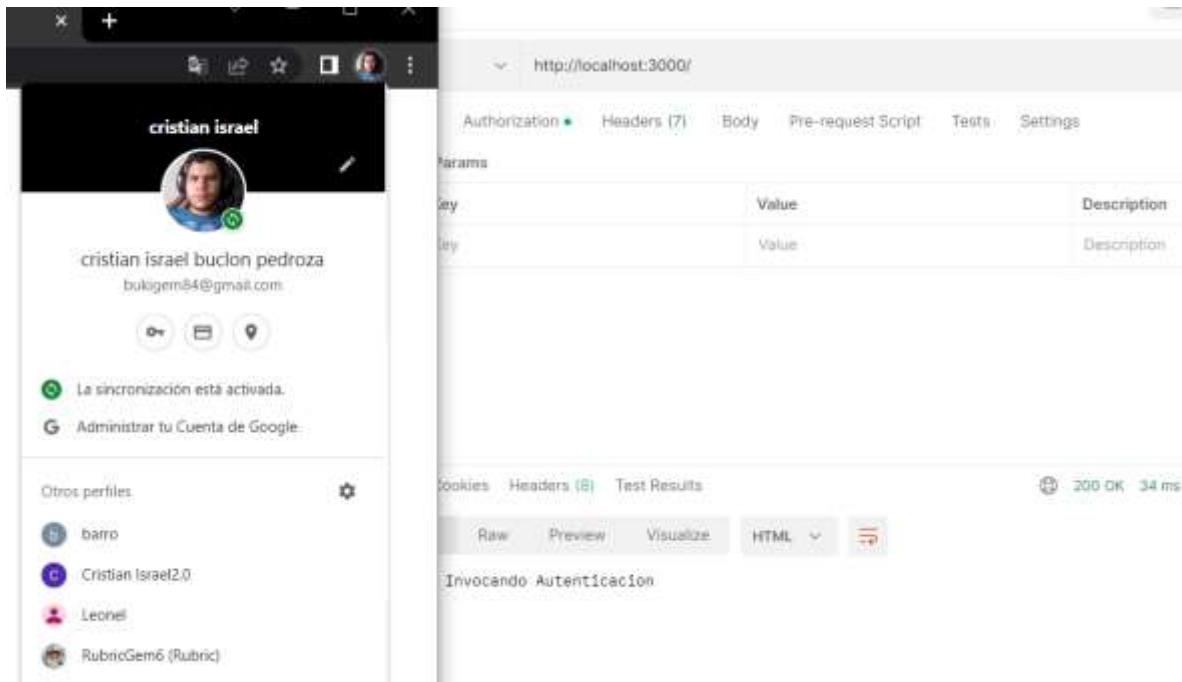
## 21. Definición de carpeta de rutas



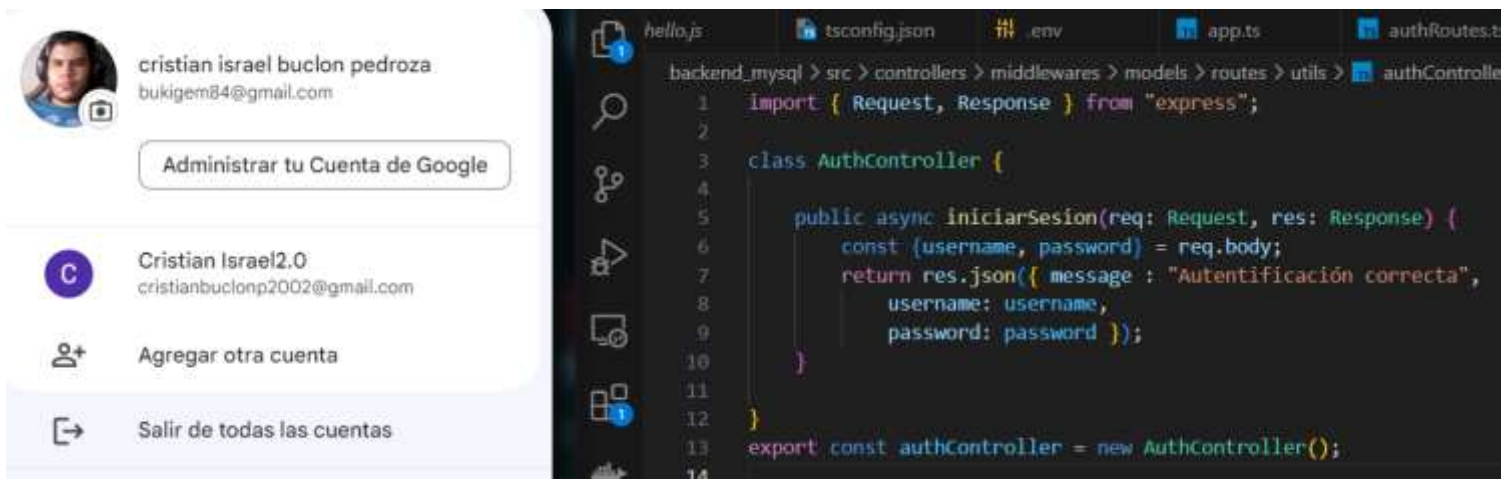
## 22. Establecimiento de archivo de rutas



## 23. Verification de la function del ruteo en Post Man

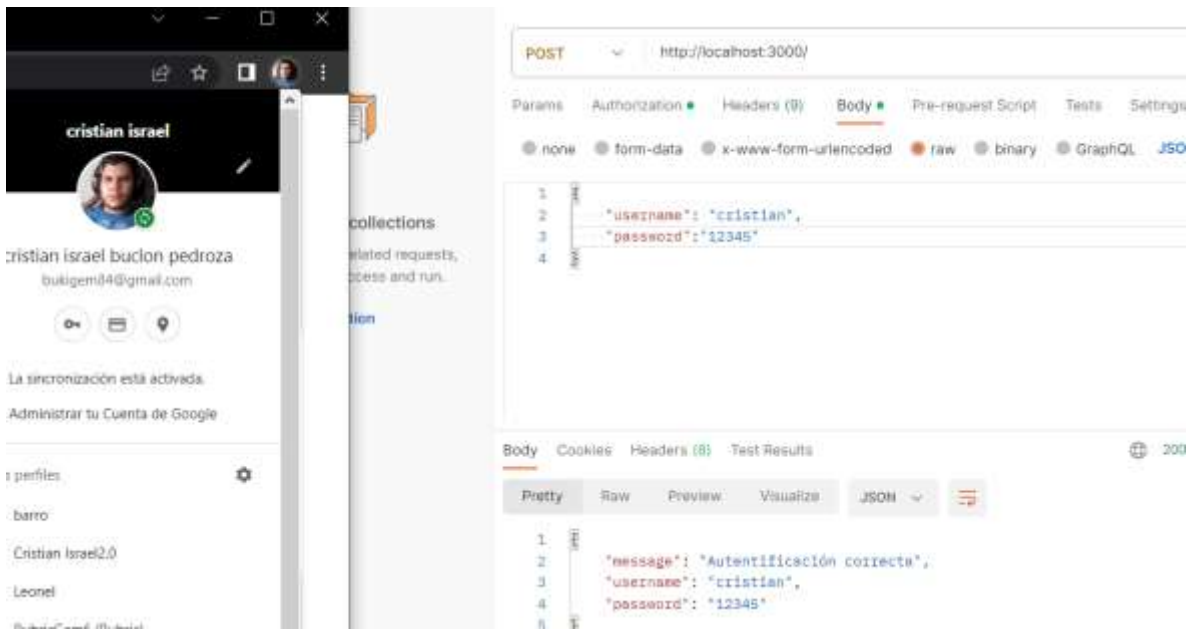


#### 24. Declaración de la clase AuthController

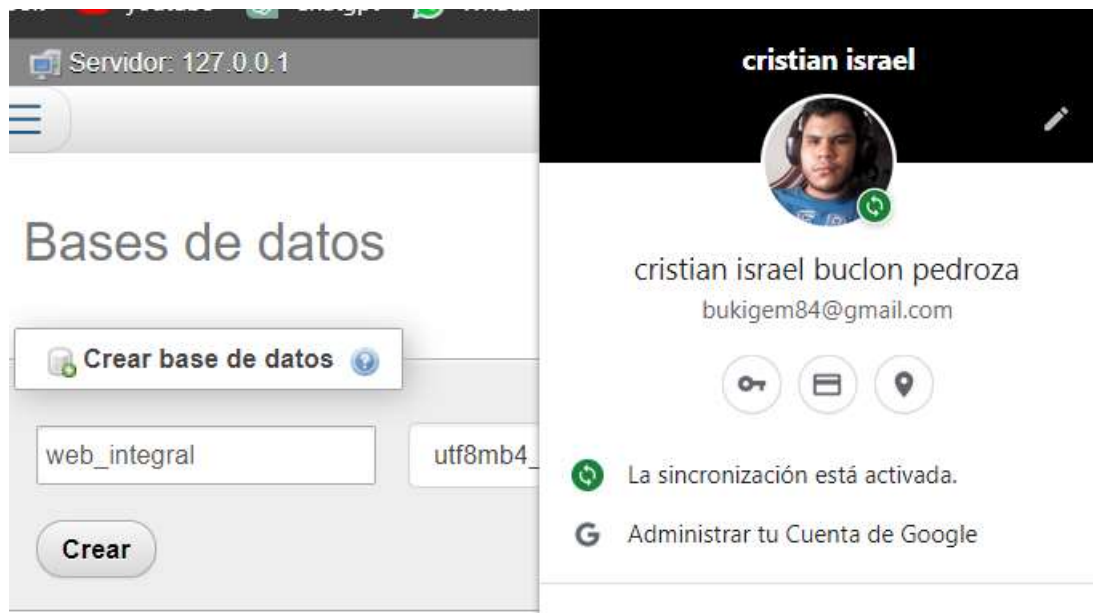


#### 25. Comprobacion de post

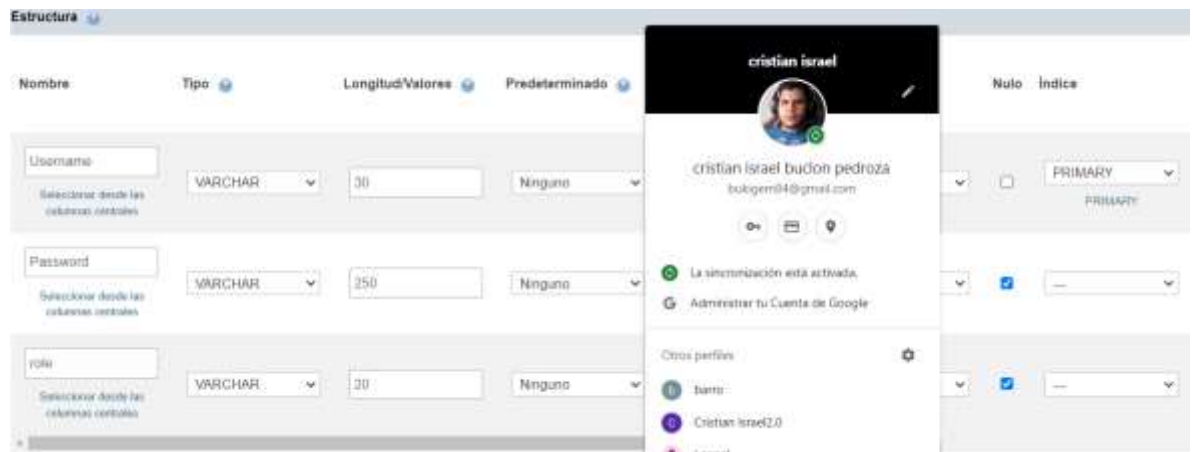




## 26. Creacion de base de datos



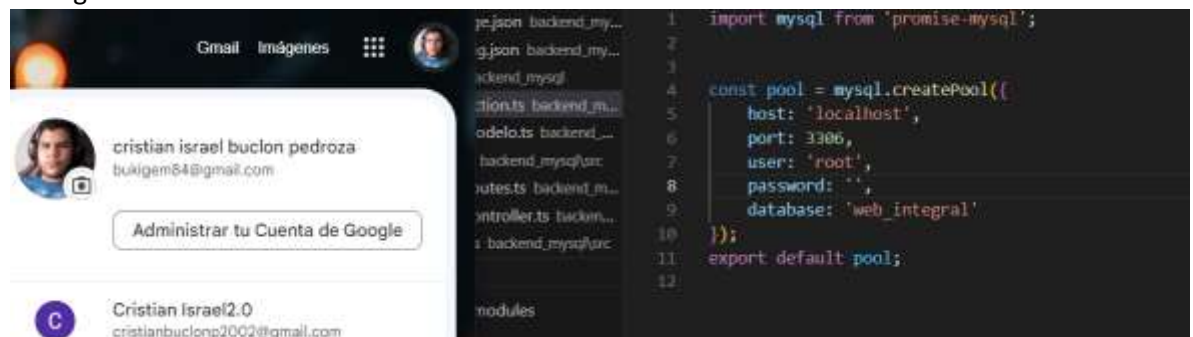
## 27. Tabla



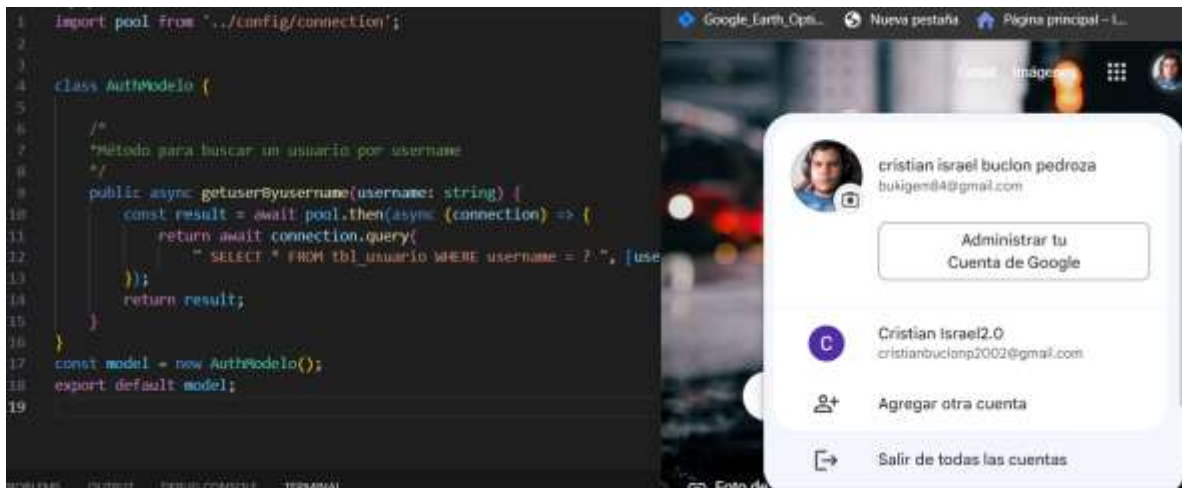
## 28. Insercion de datos



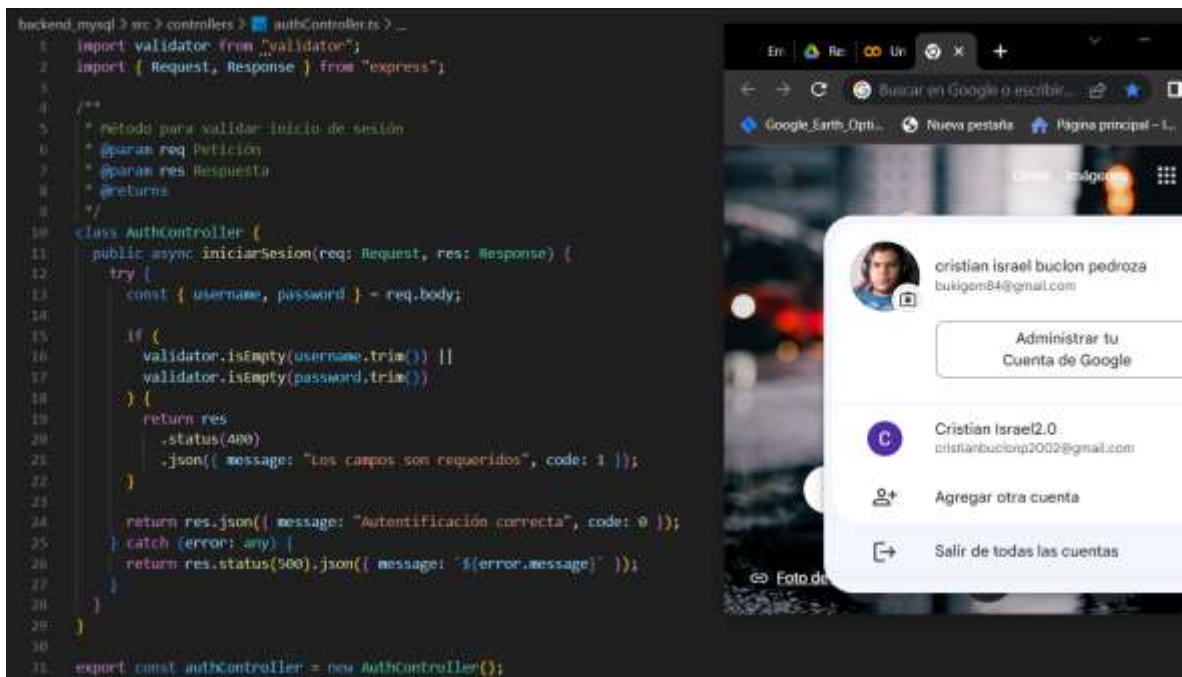
## 29. Configuracion de la conexion de la base de datos



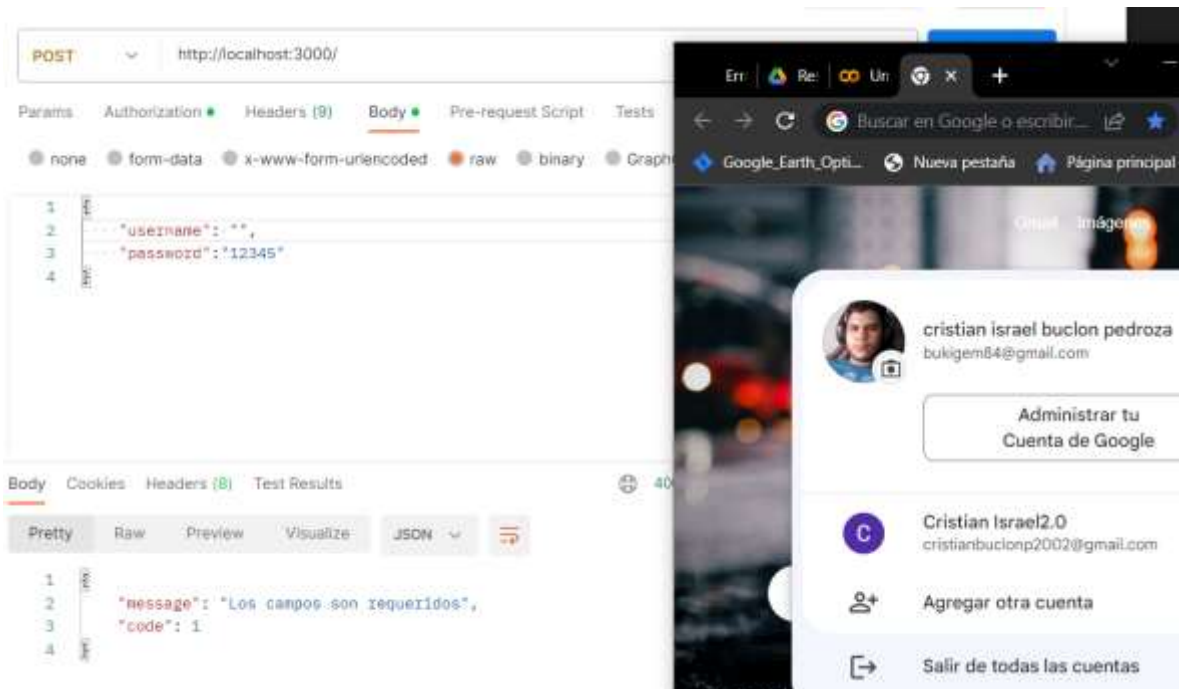
## 30. Definicion del modelo autentificacion



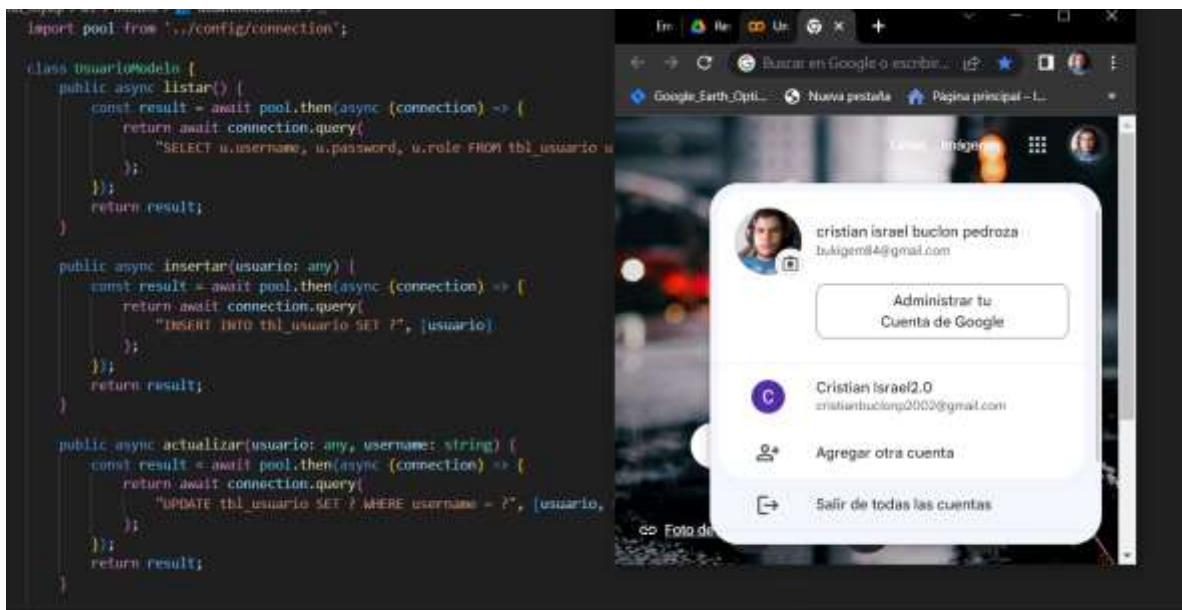
### 31. Modificación del authcontroller



### 32. Verificación de postman los campos son requeridos

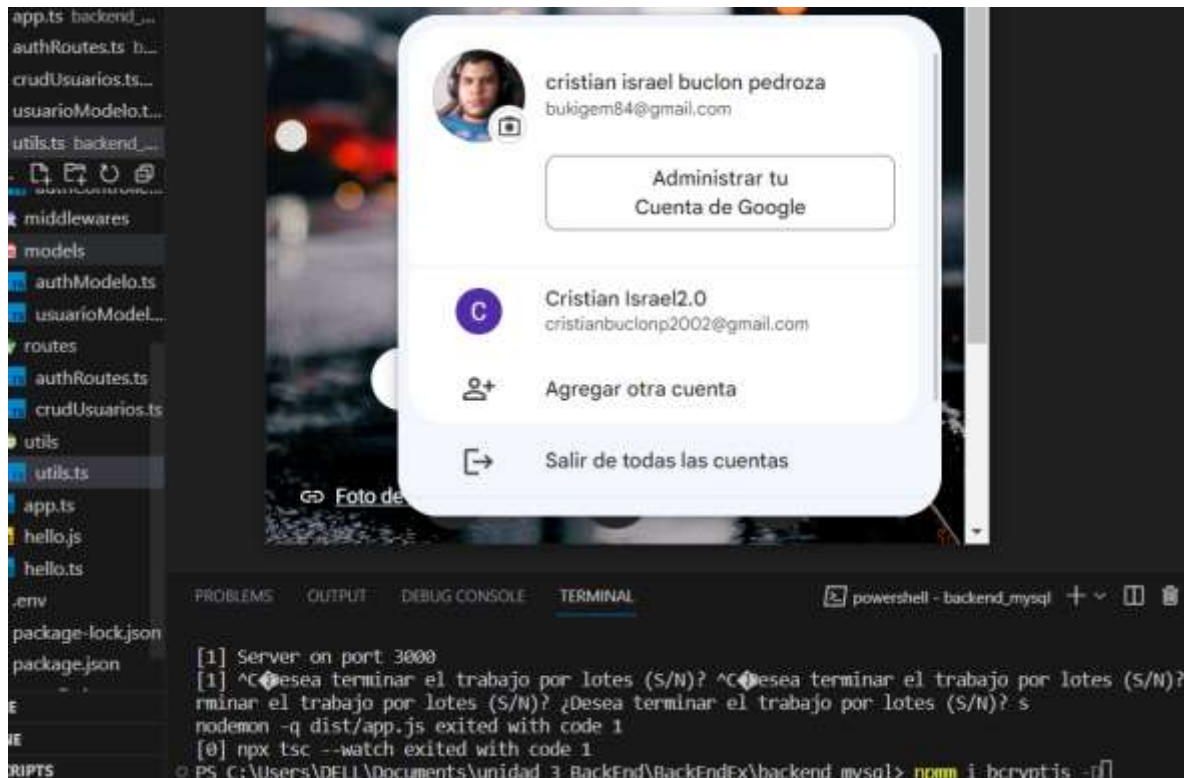


### 33. Configuración del modelo del usuario



### 34. Creación de carpeta útil e inst

### 35. Comprobación del token

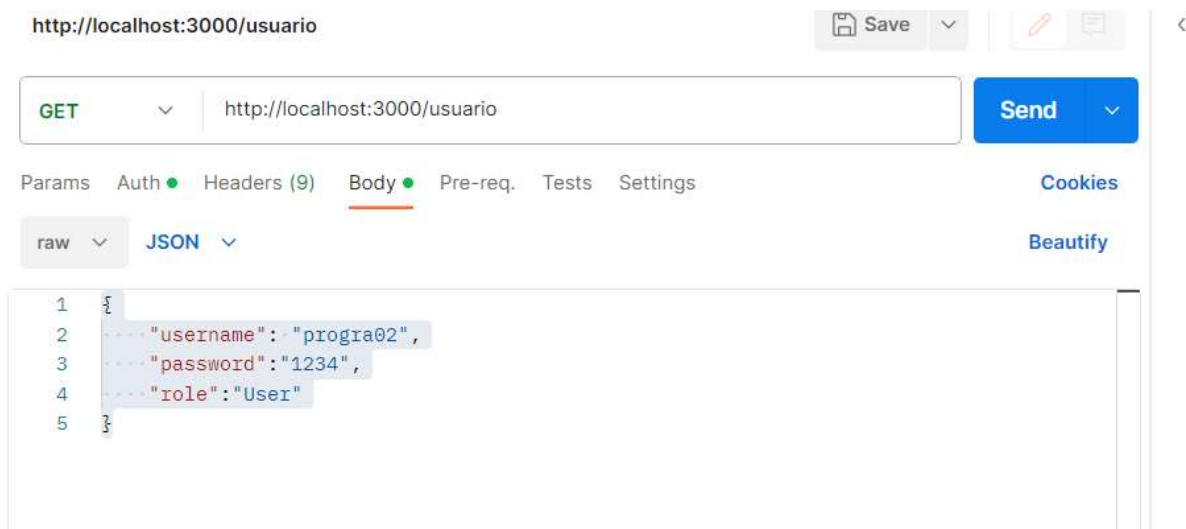


### 35- Utils





### 36— get /usuario



http://localhost:3000/usuario

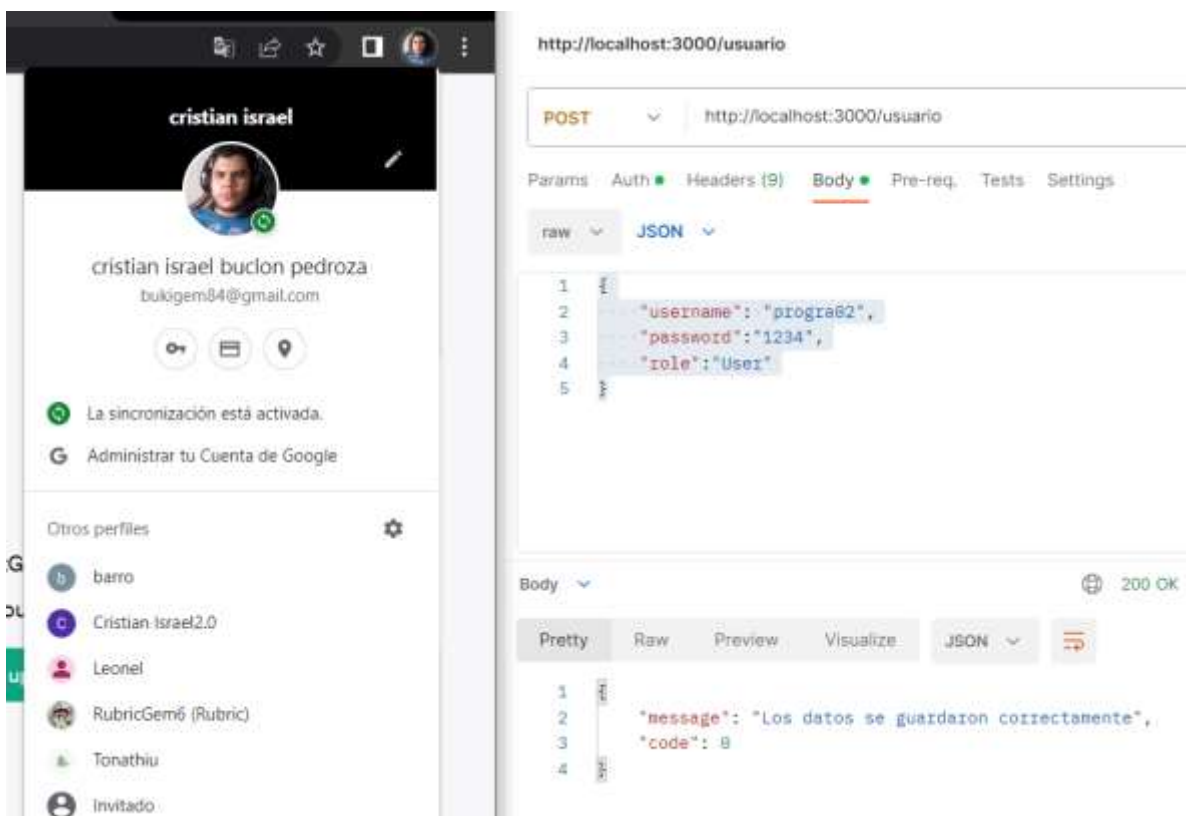
GET http://localhost:3000/usuario

Params Auth Headers (9) Body Pre-req. Tests Settings

raw JSON

```
1 {
2   "username": "progra02",
3   "password": "1234",
4   "role": "User"
5 }
```

### 37.- Post usuarios



http://localhost:3000/usuario

POST http://localhost:3000/usuario

Params Auth Headers (9) Body Pre-req. Tests Settings

raw JSON

```
1 {
2   "username": "progra02",
3   "password": "1234",
4   "role": "User"
5 }
```

Body 200 OK

Pretty Raw Preview Visualize JSON

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

Mobile App Interface:

cristian israel

cristian israel buclon pedroza

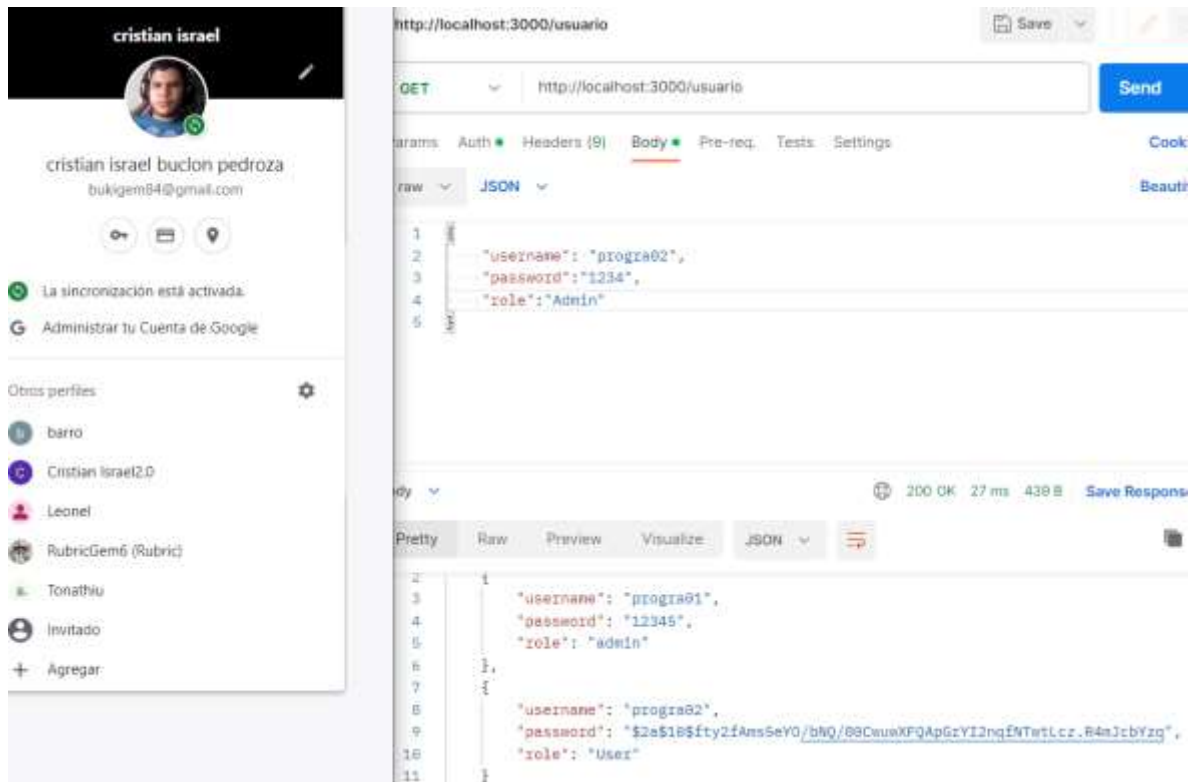
bulogem84@gmail.com

La sincronización está activada.

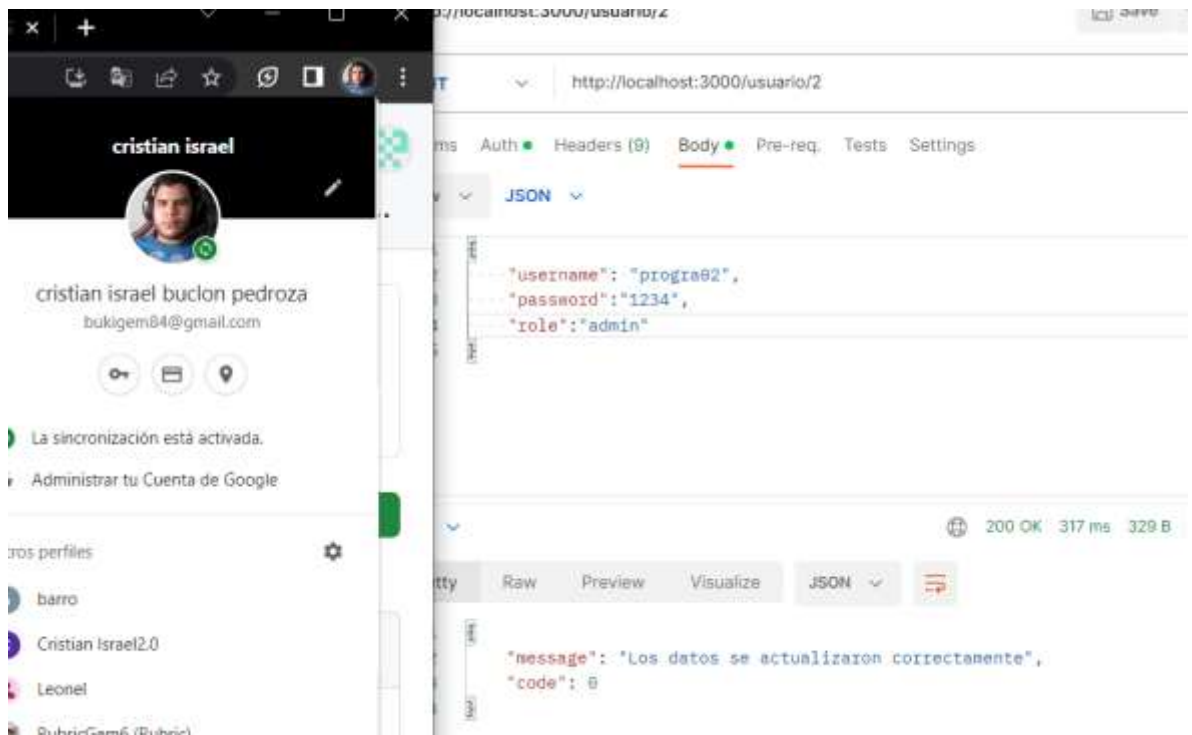
Administrar tu Cuenta de Google

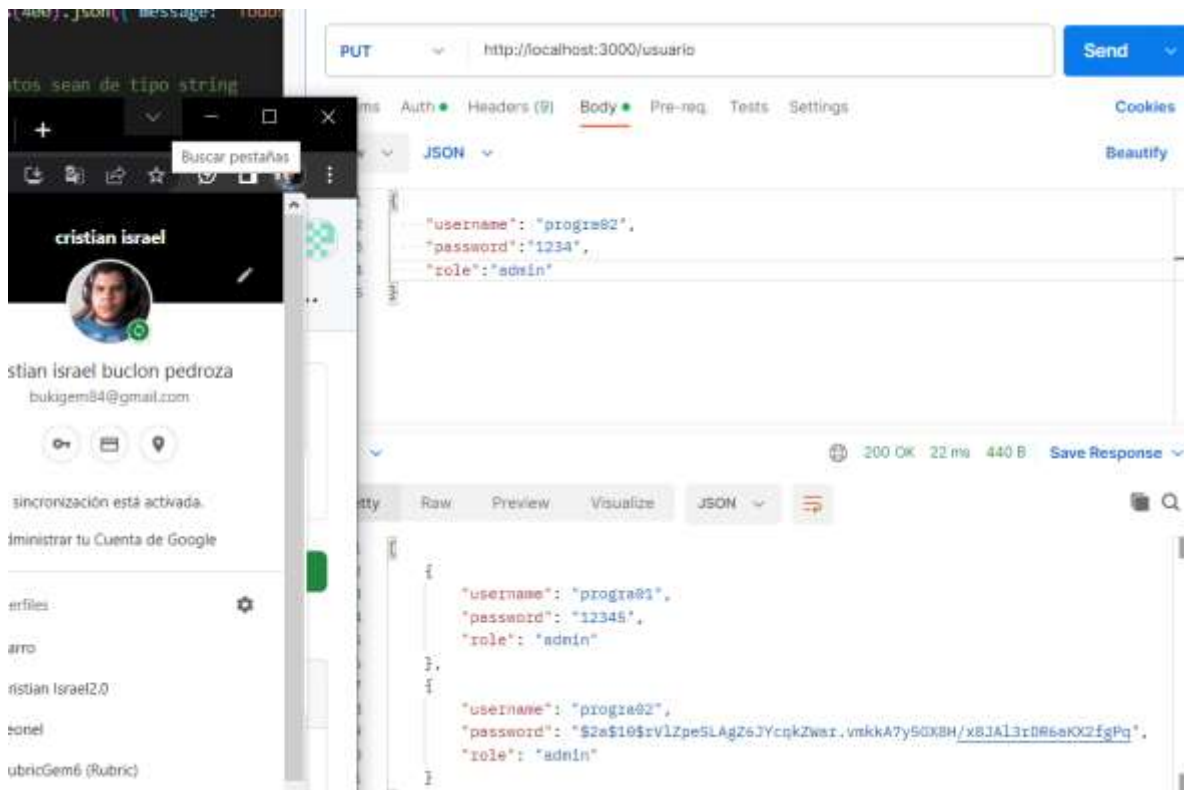
Otros perfiles

- barro
- Cristian-Israel2.0
- Leonel
- RubricGem6 (Rubric)
- Tonathiu
- Invitado



### 38. Put Usuarios





39. Delete



The screenshot shows a WhatsApp profile page. At the top, the name 'cristian israel' is displayed in bold. Below it is a circular profile picture of a man with a beard and a green checkmark icon. Under the picture, the full name 'cristian israel budon pedroza' and the email address 'budonjemi4@gmail.com' are shown. Below the email are three icons: a plus sign, an envelope, and a location pin. At the bottom, there are two status messages: 'La sincronización está activada.' with a green checkmark icon, and 'Administrar tu Cuenta de Google' with a 'G' icon. The bottom of the screen shows a section titled 'Otros perfiles' with a gear icon on the right.

[illegible]