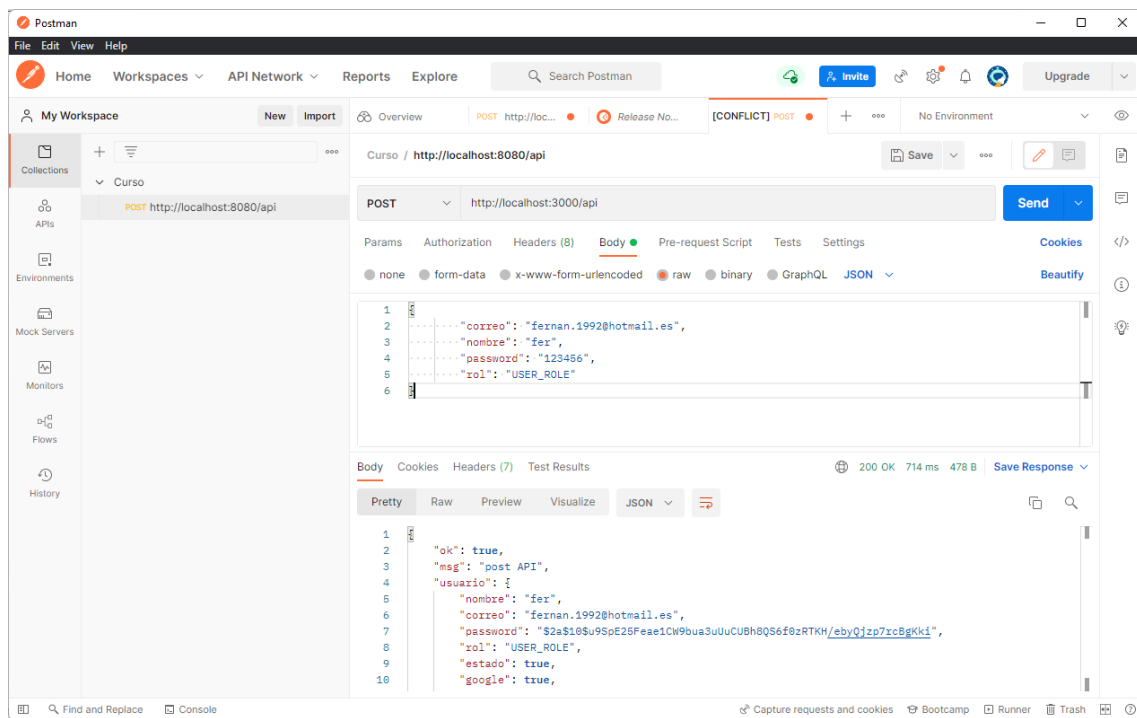
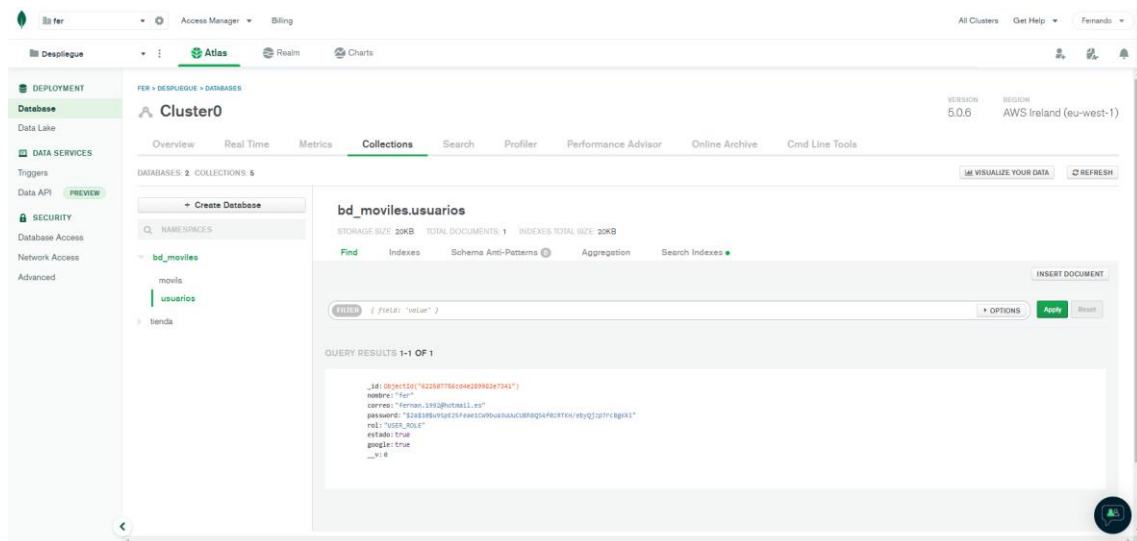


# Token

Creamos un usuario.



Verificamos si est en la db:



Instalamos la librera: (en amarillo)

```
server.js - CrudMovilesNode-main - Visual Studio Code

EXPLORADOR
CRUDMOVILESNODE-MAIN
  database
  JS config.js
  helpers
  middleware
  JS validar-JWT.js
  modelo
  JS moviles.js
  JS rol.js
  JS server.js
  JS usuario.js
  node_modules
  .env
  JS app.js
  CrudMoviles.pdf
  package-lock.json
  package.json

modelo > JS server.js > ...
1 const express = require('express');
2 require('dotenv').config();
3 const bcryptjs = require('bcryptjs');
4 const { dbConnection } = require('../database/config');
5 const Movil = require('../moviles');
6 const { body, validationResult, check } = require('express-validator');
7 const { validarJWT } = require('../middleware/validar-JWT');
8 const port = process.env.PORT;
9 const Usuario = require('../usuario');
10 const { generarJWT } = require("../helpers/generarJWT");
11 class Server {
12   constructor() {
13     this.app = express();
14     this.conectarDB();
15     this.middlewares();
16     this.rutas();
17   }
18   middlewares() {
19     this.app.use(express.json()); //Middleware para leer json;
20     this.app.use(express.static('public'));
21     //Middleware para servir la carpeta public
22   }
23   async conectarDB() {
24     await dbConnection();
25   }
26 }
27 const server = new Server();
28 server.constructor();
29 server.app.listen(port, () => {
30   console.log(`Escuchando el puerto 3000`);
31 });
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

oute.js:137:13) at middleware (C:\Users\ferna\Desktop\CrudMovilesNode-main\node\_modules\express-validator\src\middlewares\check.js:16:13) at processTicksAndRejections (node:internal/process/task\_queues:96:5) [nodemon] restarting due to changes... [nodemon] starting `node app.js` Escuchando el puerto 3000 Base de datos ONLINE PS C:\Users\ferna\Desktop\CrudMovilesNode-main> npm i install jsonwebtoken

Generamos la carpeta helpers para generar el token

```
generarJWT.js - CrudMovilesNode-main - Visual Studio Code

EXPLORADOR
CRUDMOVILESNODE-MAIN
  database
  JS config.js
  helpers
  JS generarJWT.js
  middleware
  JS validar-JWT.js
  modelo
  JS moviles.js
  JS rol.js
  JS server.js
  JS usuario.js
  node_modules
  .env
  JS app.js
  CrudMoviles.pdf
  package-lock.json
  package.json

helpers > JS generarJWT.js > ...
1 const jwt = require('jsonwebtoken');
2 const generarJWT = ( id = '' ) => {
3   return new Promise( ( resolve, reject ) => {
4     const payload = { id };
5
6     jwt.sign( payload, process.env.SECRETORPRIVATEKEY, {
7       expiresIn: '4h'
8     }, ( err, token ) => {
9       if ( err ) {
10        console.log(err);
11        reject( 'No se pudo generar el token' );
12      } else {
13        resolve( token );
14      }
15    } )
16  } )
17 }
18 module.exports = {
19   generarJWT
20 }
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

oute.js:137:13) at middleware (C:\Users\ferna\Desktop\CrudMovilesNode-main\node\_modules\express-validator\src\middlewares\check.js:16:13) at processTicksAndRejections (node:internal/process/task\_queues:96:5) [nodemon] restarting due to changes... [nodemon] starting `node app.js` Escuchando el puerto 3000 Base de datos ONLINE PS C:\Users\ferna\Desktop\CrudMovilesNode-main> npm i install jsonwebtoken

Se importa en el servidor:

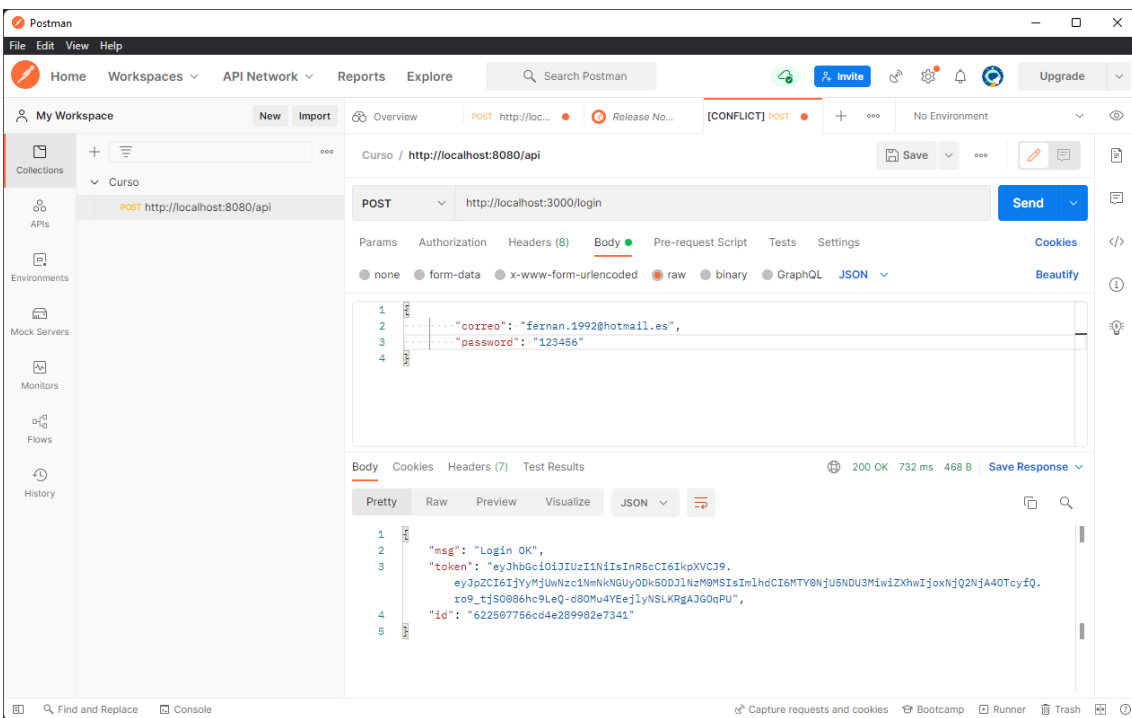
The screenshot shows the Visual Studio Code interface. On the left, the 'EXPLORADOR' (Explorer) sidebar shows the project structure for 'CRUDMOVILESNODE-MAIN', including files like 'database', 'config.js', 'helpers', 'generarJWT.js', 'middleware', 'validar-JWT.js', 'modelo', 'moviles.js', 'rol.js', 'server.js', 'usuario.js', 'node\_modules', '.env', 'app.js', 'CrudMoviles.pdf', 'package-lock.json', and 'package.json'. The 'server.js' file is selected and open in the main editor. The code in 'server.js' is as follows:

```

1  const express = require('express');
2  require('dotenv').config();
3  const bcryptjs = require('bcryptjs');
4  const { dbConnection } = require('../database/config')
5  const Movil = require('./moviles');
6  const { body, validationResult, check } = require('express-validator');
7  const { validarJWT } = require('../middleware/validar-JWT');
8  const port = process.env.PORT;
9  const Usuario = require('./usuario');
10 const { generarJWT } = require('../helpers/generarJWT');
11 class Server {
12   constructor() {
13     this.app = express();
14     this.conectarDB();
15     this.middlewares();
16     this.rutas();
17   }
18   middlewares() {
19     this.app.use(express.json()); //Middleware para leer json;
20     this.app.use(express.static('public'));
21     //^Middleware para servir la carpeta public
22   }
23   async conectarDB() {
24     await dbConnection()
  
```

The terminal at the bottom shows the command 'npm i install jsonwebtoken' being executed, with the output indicating that the package is being installed.

Comprobamos el login:





Se importa

```
const { body, validationResult, check } = require('express-validator');
const { validarJWT } = require('../middlewares/validar-jwt');
const { validarToken } = require('../middlewares/validar-token');
```

Y se utiliza en el borrado:

```
const id = req.params.id;
let { password, ...resto } = req.body;
//**** le hago el hash a la contraseña ****
const salt = bcryptjs.genSaltSync();
password = bcryptjs.hashSync(password, salt);
resto.password = password;
await Usuario.findByIdAndupdate(id, resto);
res.status(403).json({
  ok: true,
  msg: 'put API',
  id,
  resto
})
})

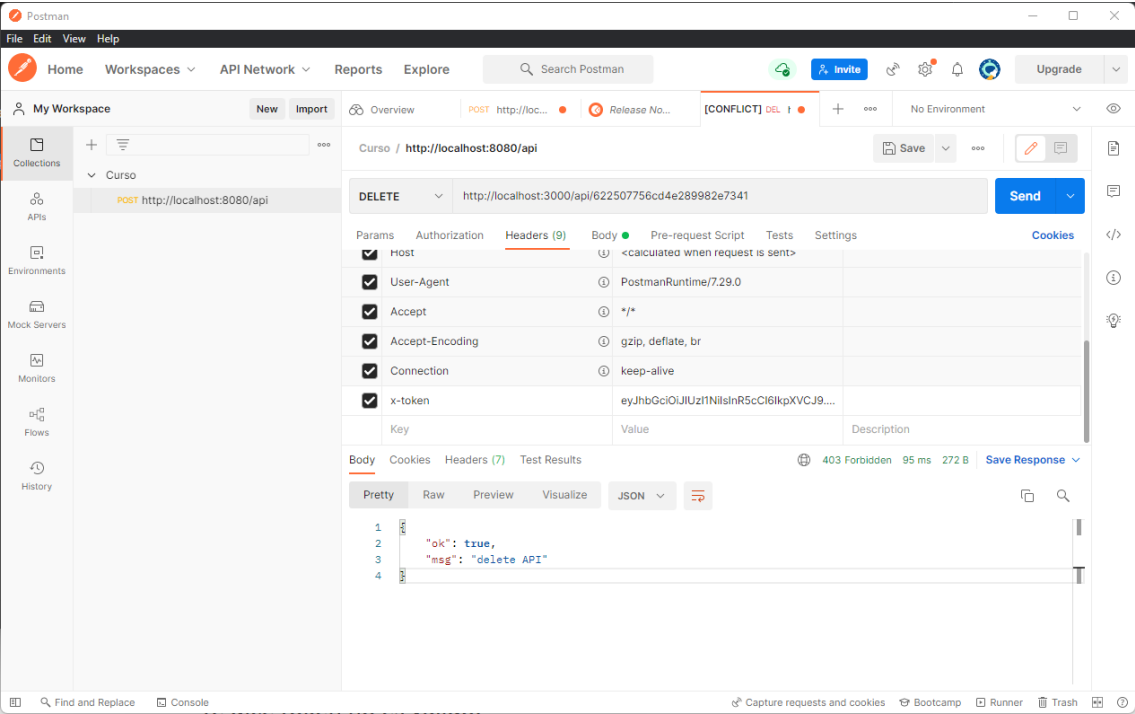
this.app.delete('/api/id', validarJWT, async function (req, res) {
  const id = req.params.id;
  await Usuario.findByIdAndDelete(id);
  res.status(403).json({
    ok: true,
    msg: 'delete API'
  })
})

this.app.get('/saludo', function (req, res) {
  res.send('¡Hola 2024c/hi!')
})

this.app.get('*', function (req, res) {
  res.sendFile(__dirname + '/404.html')
})

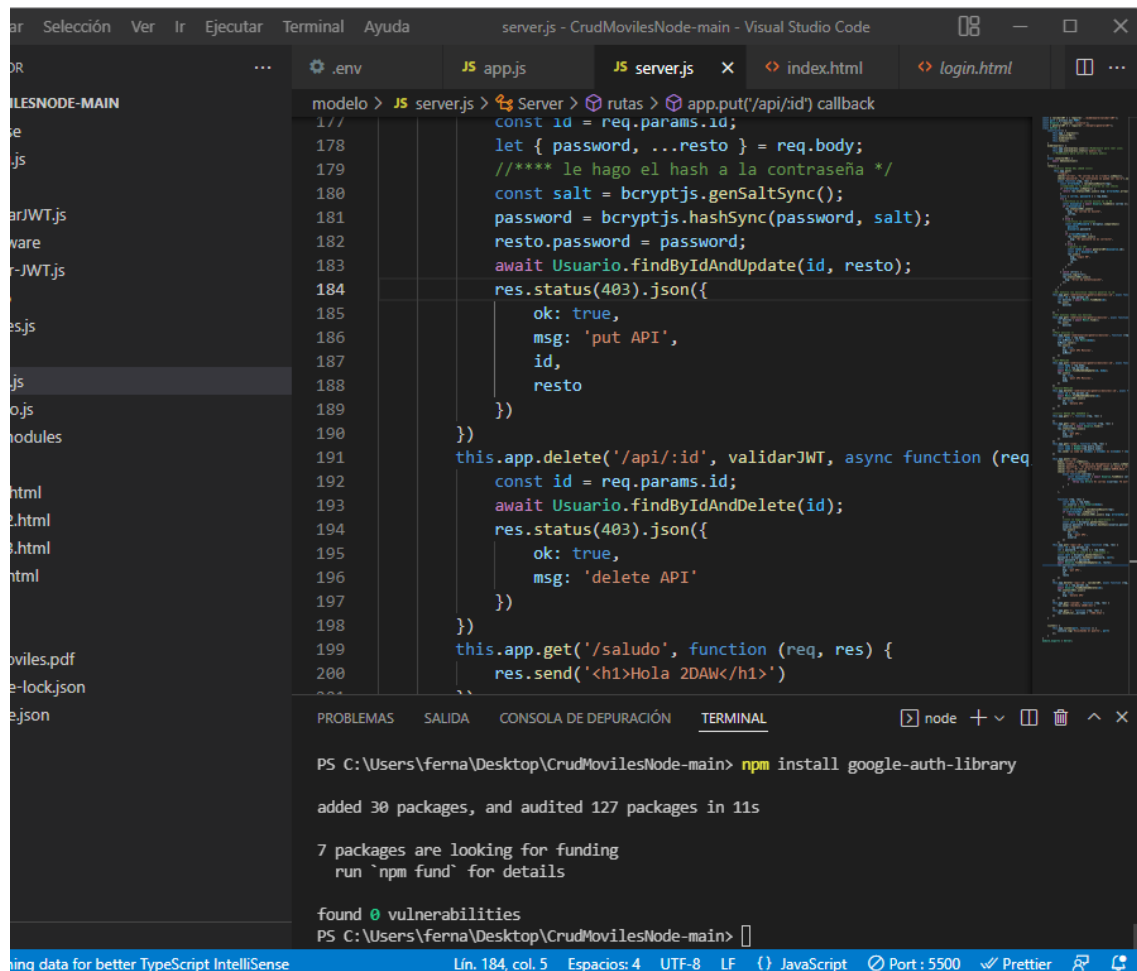
listen() {
  this.app.listen(port, function () {
    console.log('Escuchando el puerto', port)
  });
}
```

Borrado con el token:



GOOGLE:

Librería:



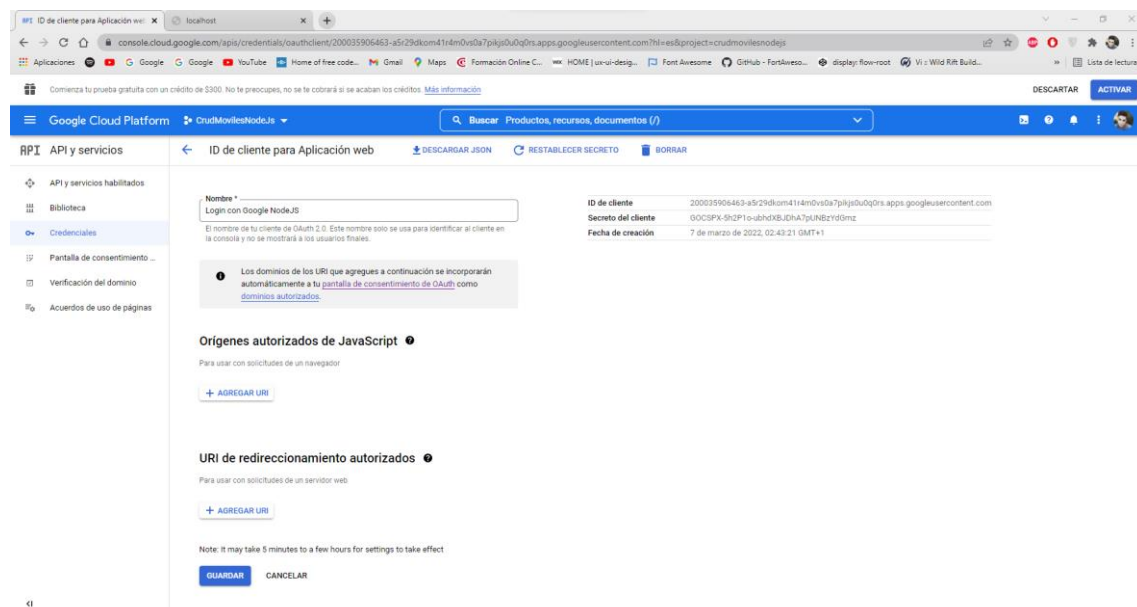
The screenshot shows a Visual Studio Code editor with a project named 'server.js - CrudMovilesNode-main'. The main file is 'server.js', which contains the following code:

```
177 //
178 const id = req.params.id;
179 let { password, ...resto } = req.body;
180 //**** le hago el hash a la contraseña ****
181 const salt = bcryptjs.genSaltSync();
182 password = bcryptjs.hashSync(password, salt);
183 resto.password = password;
184 await Usuario.findByIdAndUpdate(id, resto);
185 res.status(403).json({
186   ok: true,
187   msg: 'put API',
188   id,
189   resto
190 })
191 this.app.delete('/api/:id', validarJWT, async function (req, res) {
192   const id = req.params.id;
193   await Usuario.findByIdAndDelete(id);
194   res.status(403).json({
195     ok: true,
196     msg: 'delete API'
197   })
198 })
199 this.app.get('/saludo', function (req, res) {
200   res.send('<h1>Hola 2DAW</h1>')
```

The terminal window at the bottom shows the following output:

```
PS C:\Users\ferna\Desktop\CrudMovilesNode-main> npm install google-auth-library
added 30 packages, and audited 127 packages in 11s
7 packages are looking for funding
run 'npm fund' for details
found 0 vulnerabilities
PS C:\Users\ferna\Desktop\CrudMovilesNode-main>
```

Añadimos los id de cliente y secreto del cliente




The screenshot shows the Google Cloud Platform console for a project named 'CrudMovilesNode.js'. The page is titled 'ID de cliente para Aplicación web' and displays the following information:

Nombre *	ID de cliente
Login con Google Node.js	200035906463-a5c29d0m414m0v0a7pikjsdu0qrs.apps.googleusercontent.com


Secreto del cliente	Fecha de creación
GOCSPK-5h2P1o-ubhdXBjDha7pUN8tYd0mz	7 de marzo de 2022, 02:43:21 GMT+1

Below the table, there is a section for 'Orígenes autorizados de JavaScript' and 'URI de redireccionamiento autorizados'. The 'Orígenes autorizados de JavaScript' section has a button '+ AGREGAR URI'. The 'URI de redireccionamiento autorizados' section has a button '+ AGREGAR URI'. At the bottom, there is a note: 'Note: it may take 5 minutes to a few hours for settings to take effect'. There are also buttons for 'GUARDAR' and 'CANCELAR'.

Login con google:


 Sign In - Google Accounts - Google Chrome

accounts.google.com/gsi/select?client\_id=200035906463-a5r29dkom4...


 Iniciar sesión con Google

## Selecciona una cuenta


para ir a CrudMovilesNodeJs




**Fernando Olmedo**  
fernando.olmedo.ortiz.1992@gmail.com



**fernando olmedo**  
narsxo92@gmail.com



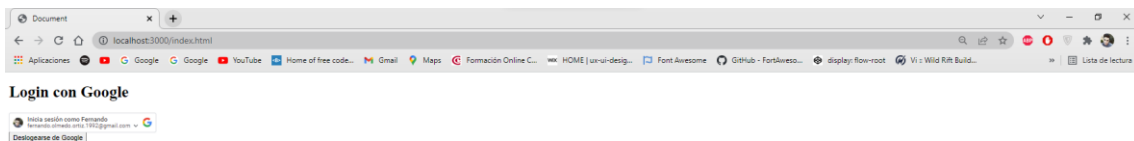
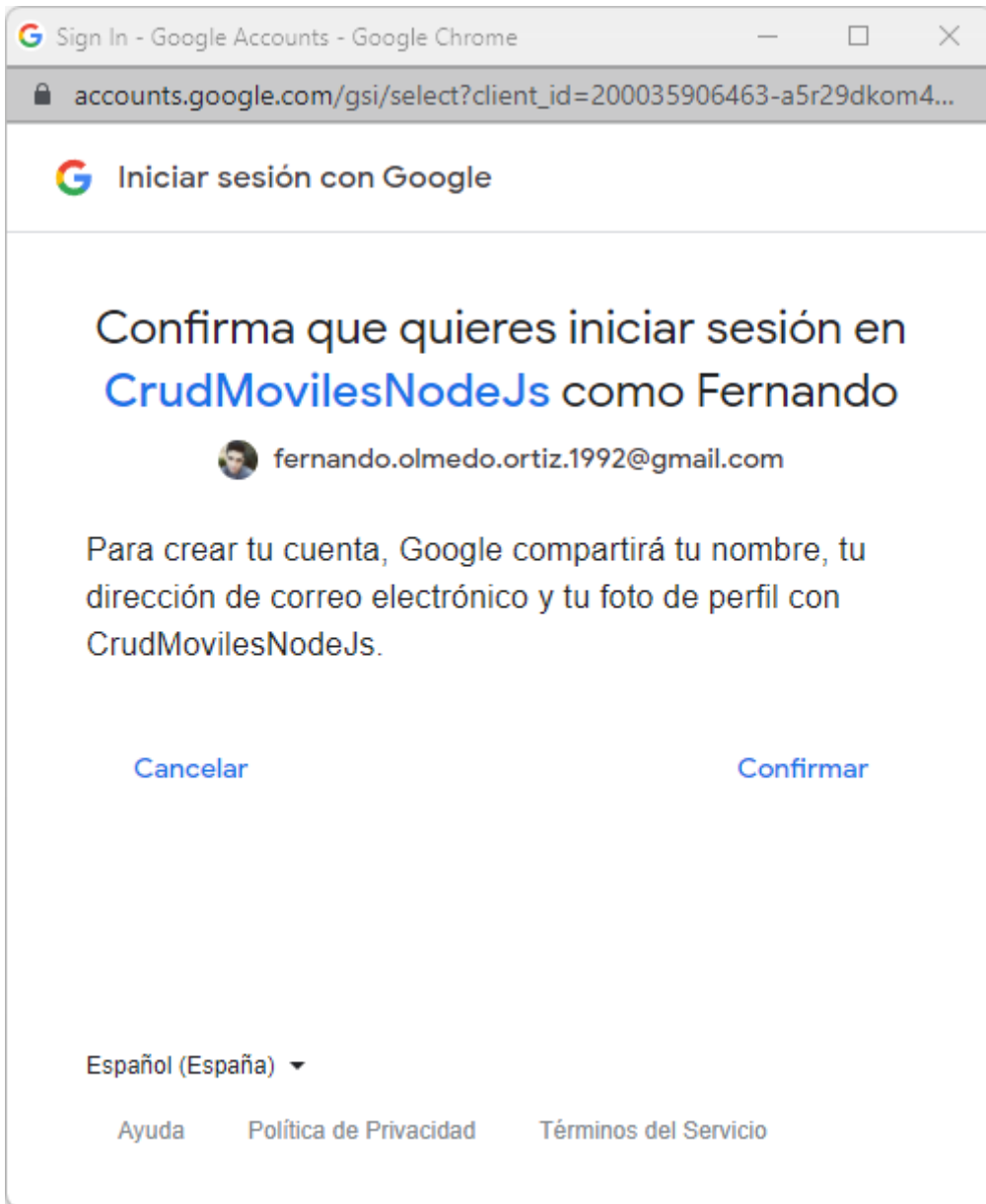
**Fernando Olmedo Rosado**  
feolmedorosado@gmail.com



Usar otra cuenta

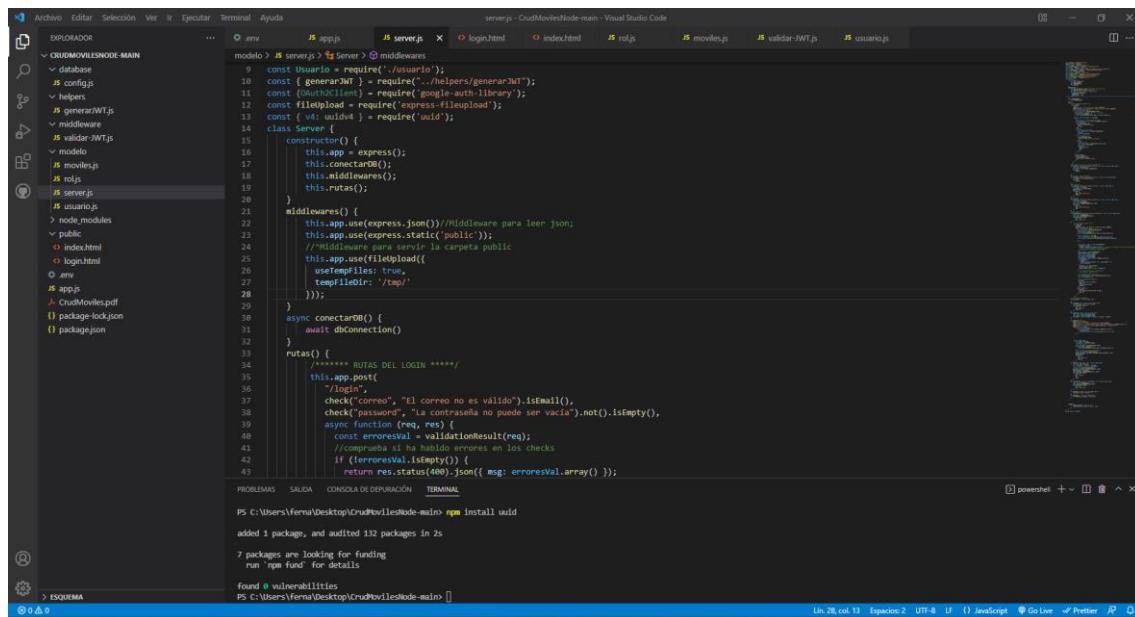
**Español (España)** ▼

[Ayuda](#) [Política de Privacidad](#) [Términos del Servicio](#)





Subida de archivos:



```
const Usuario = require('../usuario');
const { generateJWT } = require('../helpers/generarJWT');
const { checkEmail } = require('google-auth-library');
const fileupload = require('express-fileupload');
const { v4: uuidv4 } = require('uuid');

class Server {
  constructor() {
    this.app = express();
    this.conectarDB();
    this.addMiddleware();
    this.rutas();
  }

  middleware() {
    this.app.use(express.json()); //Middleware para leer json;
    this.app.use(express.static('public'));
    //Middleware para servir la carpeta public
    this.app.use(fileupload({
      useTempFiles: true,
      tempFileDir: '/tmp/'
    }));
  }

  async conectarDB() {
    await dbConnection();
  }

  rutas() {
    //***** RUTAS DEL LOGIN *****/
    this.app.post(
      '/login',
      check('correo', 'El correo no es válido').isEmail(),
      check('password', 'La contraseña no puede ser vacía').not().isEmpty(),
      async function (req, res) {
        const erroresVal = validationResult(req);
        //Comprobar si ha habido errores en los checks
        if (erroresVal.isEmpty()) {
          return res.status(400).json({ msg: erroresVal.array() });
        }
      }
    );
  }
}
```

PS C:\Users\Fernando\Desktop\CrudMovieNode-main> npm install uuid

added 1 package, and audited 132 packages in 2s

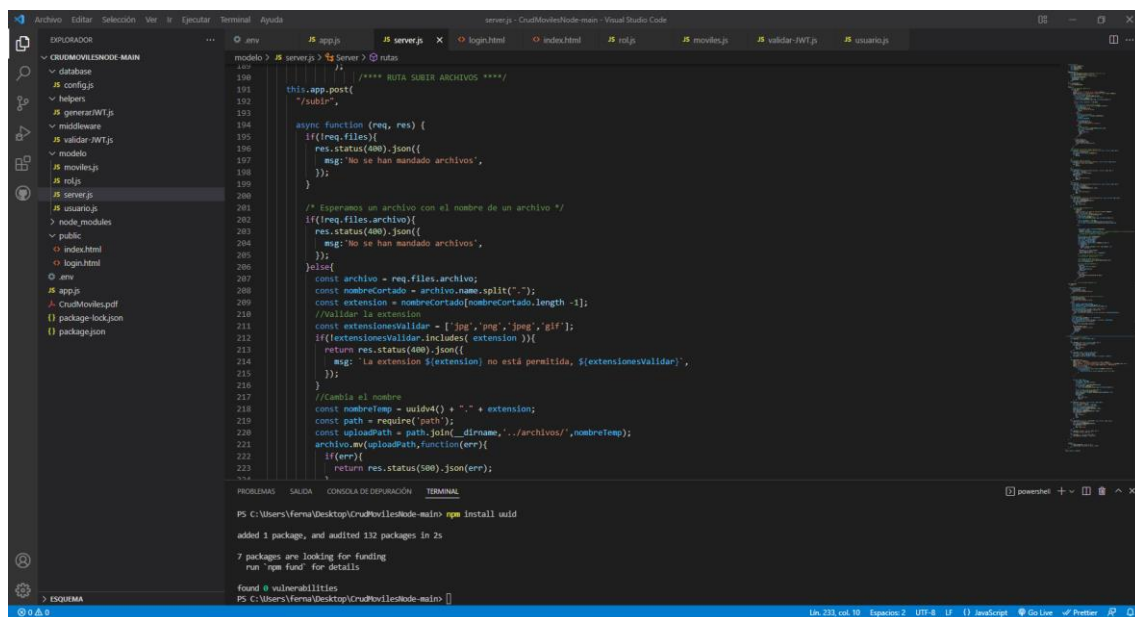
7 packages are looking for funding

run 'npm fund' for details

found 0 vulnerabilities

-se añade la variable fileupload y uuidv4 además de incorporar el middleware de fileupload.

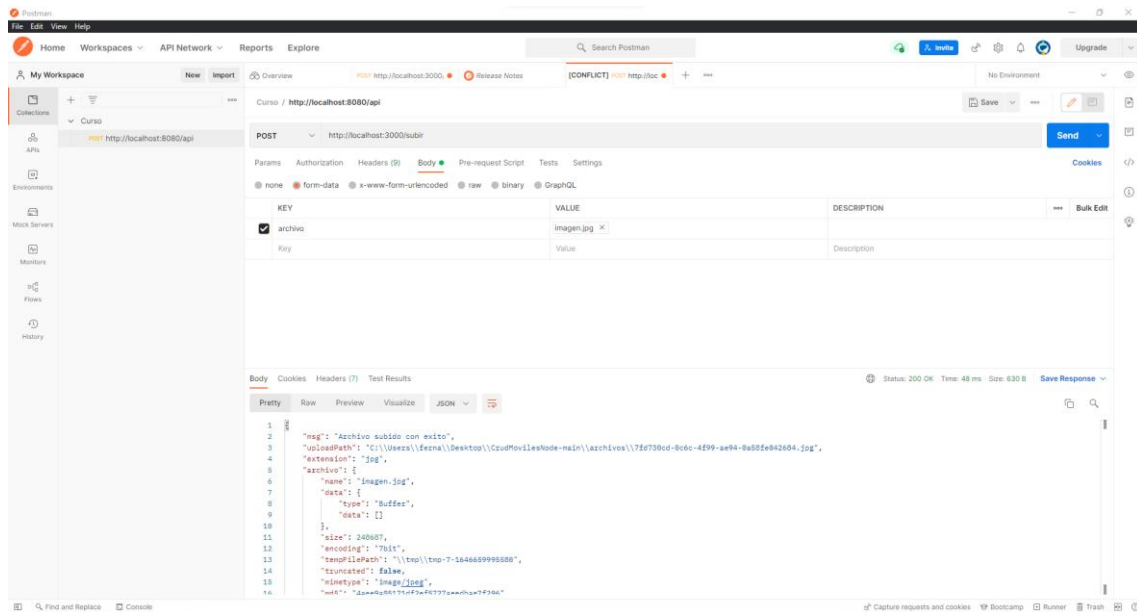
Añadimos el método post de añadir las imágenes:



```
// ***** RUTA SUBIR ARCHIVOS *****
this.app.post(
  '/subir',
  async function (req, res) {
    if (!req.files) {
      res.status(400).json({
        msg: 'No se han mandado archivos',
      });
    }

    // Esperamos un archivo con el nombre de un archivo
    if (req.files.archivo) {
      res.status(400).json({
        msg: 'No se han mandado archivos',
      });
    } else {
      const archivo = req.files.archivo;
      const nombreCortado = archivo.name.split('.');
      const extension = nombreCortado[nombreCortado.length - 1];
      //Validar la extensión
      const extensionesValidar = ['.jpg', '.png', '.jpeg', '.gif'];
      if (!extensionesValidar.includes(extension)) {
        return res.status(400).json({
          msg: 'La extensión ' + extension + ' no está permitida, ' + extensionesValidar,
        });
      }
      //Cambia el nombre
      const nombreTemp = uuidv4() + '.' + extension;
      const path = require('path');
      const uploadPath = path.join(__dirname, '..', 'archivos/', nombreTemp);
      archivo.mv(uploadPath, function (err) {
        if (err) {
          return res.status(500).json(err);
        }
      });
    }
  }
);
```

Archivo subido con éxito:



Le cambia el nombre por si se duplica

