

```
<!--Programación IV-->
```

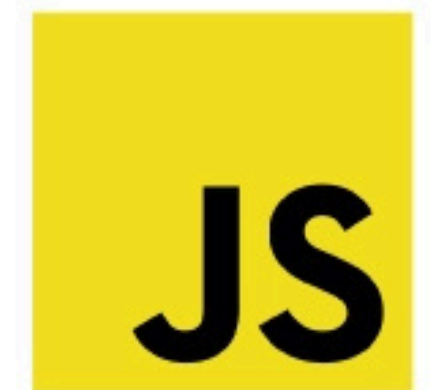
```
Express.js { El  
framework minimalista  
para Node.js
```

```
}
```

```
<Creado por="Grupo 3"/>
```



Express



Contenidos



- 01 Qué es Express.js?
- 02 Instalación y configuración.
- 03 Creación de un servidor con Express.
- 04 Diferencias entre una aplicación con Node.js puro y una con Express.
- 05 Demostración práctica

1. ¿Qué es Express.js y por qué se usa en desarrollo Web? {

Es un entorno para aplicaciones web que al ser basado en Node.js aprovecha de sus capacidades, como su modelo de entrada/salida no bloqueante, lo que lo hace ideal para aplicaciones escalables y en tiempo real.

Express.js es ampliamente utilizado en desarrollo web por diversas razones:

- Escalabilidad y rendimiento.
- Flexibilidad y personalización.
- Simplificación de tareas comunes.
- Rapidez de desarrollo.
- Amplio soporte de la comunidad.

}

```
<!--Resumen: herramienta poderosa y flexible que aprovecha al máximo las capacidades de Node.js, haciendo que sea una excelente elección para construir aplicaciones web rápidas, escalables y adaptadas a necesidades específicas-->
```

2. Instalación y configuración básica de Express. PASO 1 {

NODE.JS

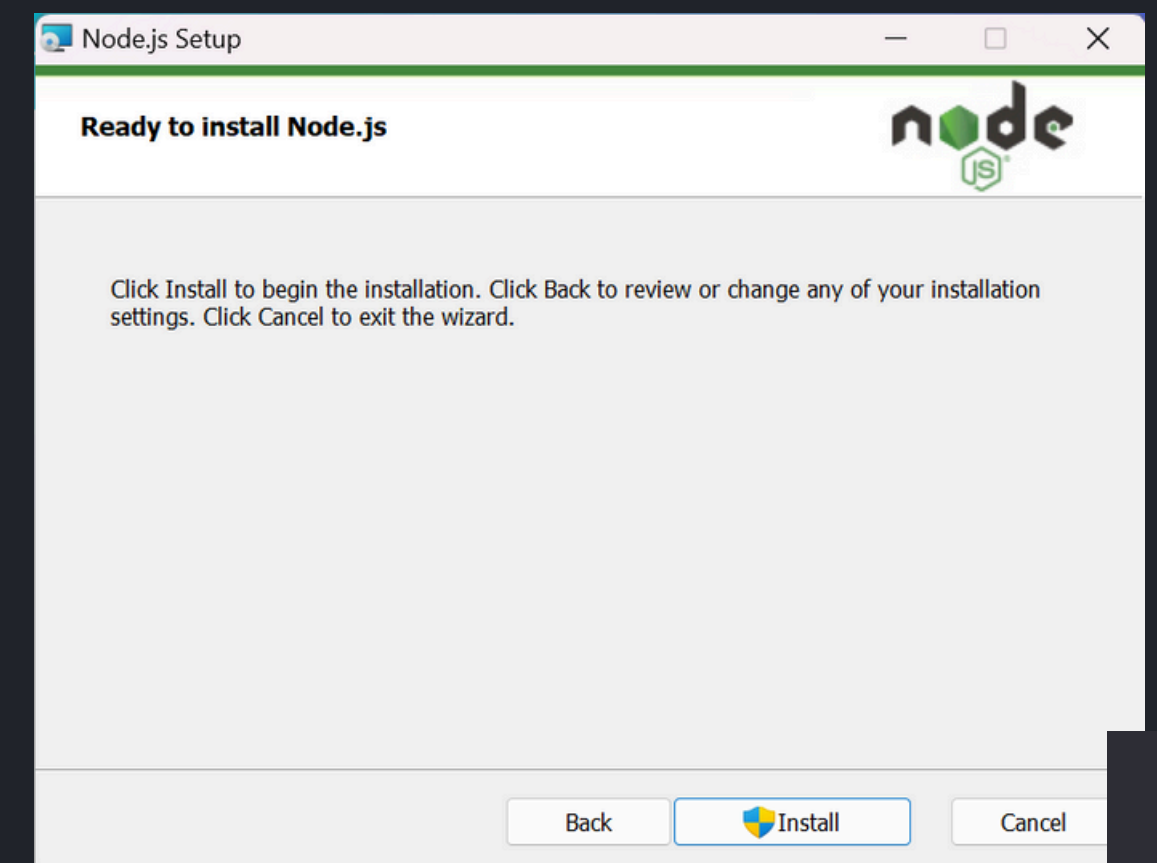
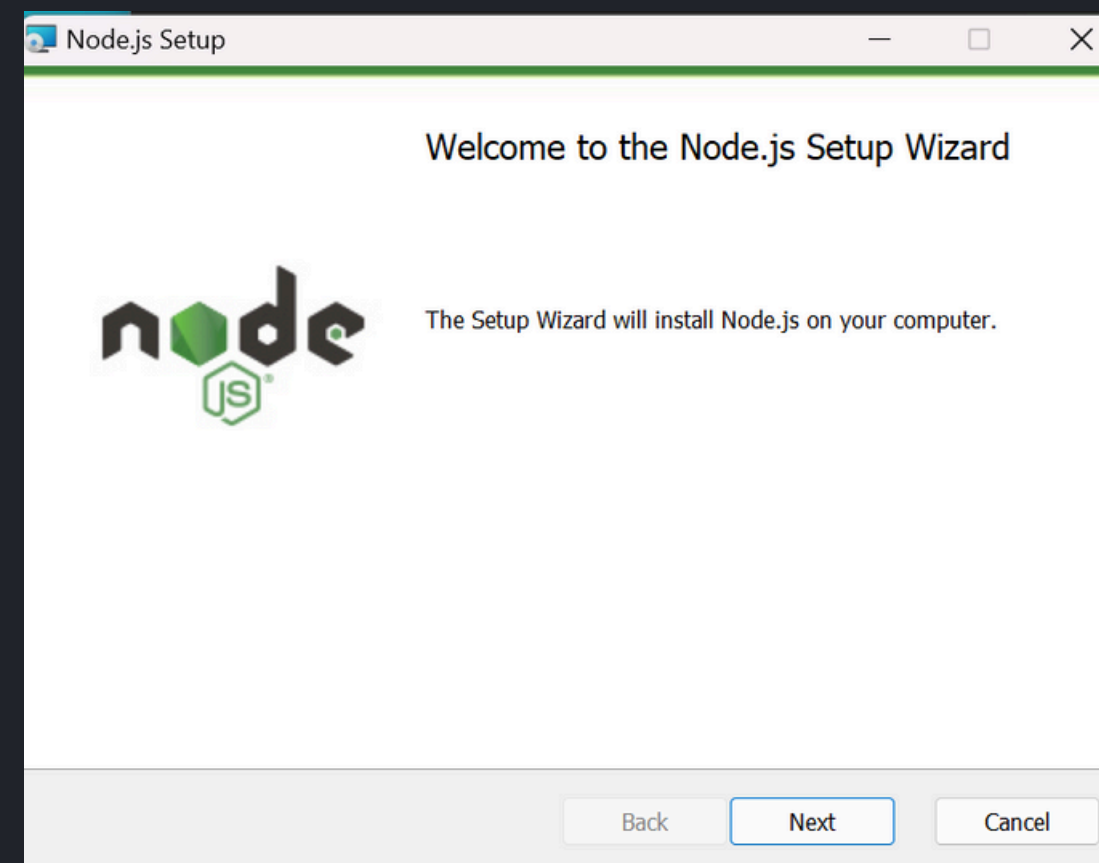
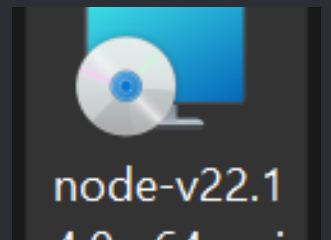
Node.js es un entorno de ejecución de JavaScript gratuito, de código abierto y multiplataforma que permite a los desarrolladores crear servidores, aplicaciones web, herramientas de línea de comandos y scripts.

<https://nodejs.org/en/downloadjs.com>

Descargar



Instalador de Windows (.msi)



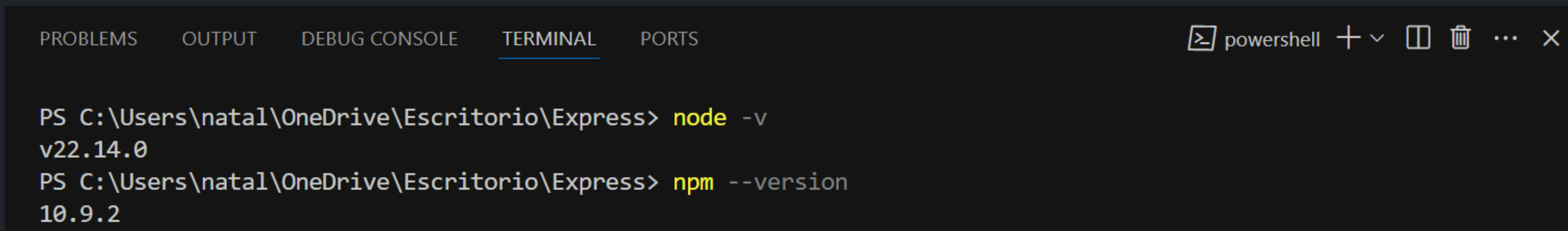
}

VERIFICANDO LA VERSION DE NODE.JS

COMANDOS EN CONSOLA (TERMINAL):

Node - v: Muestra las versiones instaladas de Node.js

Npm --version: Muestra la version instalada del gestor de paquetes que viene incluido con Node.js



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + v [icon] [icon] ... X

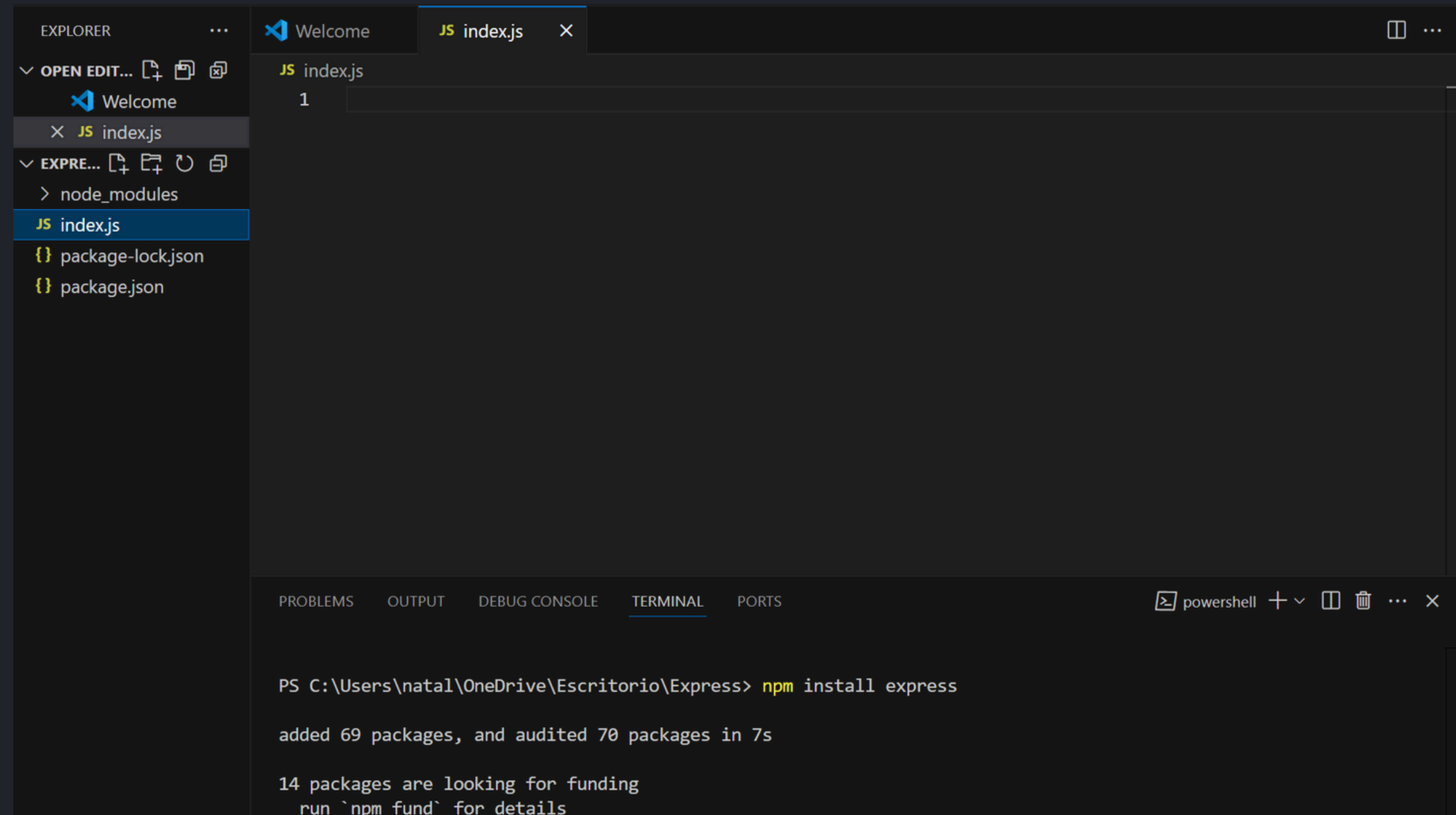
PS C:\Users\natal\OneDrive\Escritorio\Express> node -v
v22.14.0
PS C:\Users\natal\OneDrive\Escritorio\Express> npm --version
10.9.2
```

}

2. Instalación y configuración básica de Express. PASO 2 {

EXPRESS.JS

Para instalar Express.js como una dependencia se utiliza el comando {npm install express}



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure: 'Welcome', 'index.js', 'node_modules', 'package-lock.json', and 'package.json'. The 'index.js' file is selected. The main editor area shows the 'index.js' file with a single line of code. The Terminal at the bottom shows the command 'npm install express' and its output: 'added 69 packages, and audited 70 packages in 7s' and '14 packages are looking for funding run `npm fund` for details'.

```
PS C:\Users\natal\OneDrive\Escritorio\Express> npm install express

added 69 packages, and audited 70 packages in 7s

14 packages are looking for funding
run `npm fund` for details
```

}

3.Creación de un servidor con Express: Middleware, Rutas y Controladores.{

El Middleware es una función intermedia que se ejecuta antes o después de una ruta específica. Su función principal es:

- Procesar solicitudes entrantes (peticiones HTTP).
- Modificar o revisar estas solicitudes.
- Realizar tareas comunes como validar datos, seguridad o imprimir registros (logs).

Las rutas determinan cómo tu aplicación responde a solicitudes HTTP específicas en ciertos endpoints (direcciones URL).

- Las rutas reciben solicitudes desde el navegador o cliente.
- Definen qué sucede cuando una dirección específica se visita.

Un Controlador es una función separada que maneja específicamente la lógica detrás de cada ruta. Se utiliza para organizar mejor tu aplicación:

- Mantiene tu código limpio y ordenado.
- Separa claramente la lógica específica de cada operación.

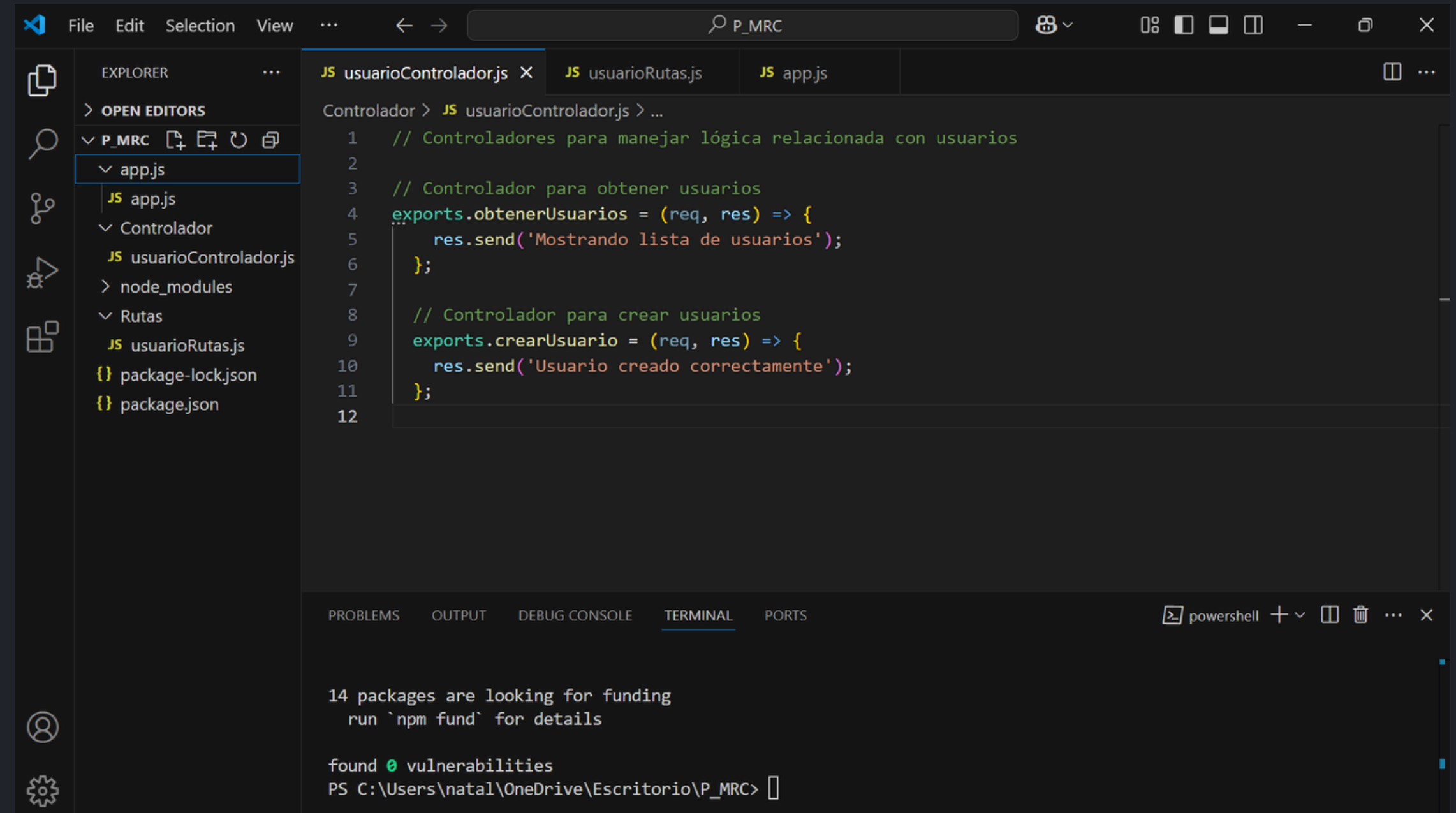
}

Se realiza una carpeta con el nombre del Proyecto

Se abre una nueva terminal y se ejecutan los comandos: NPM INIT -Y, NPM INSTALL EXPRESS

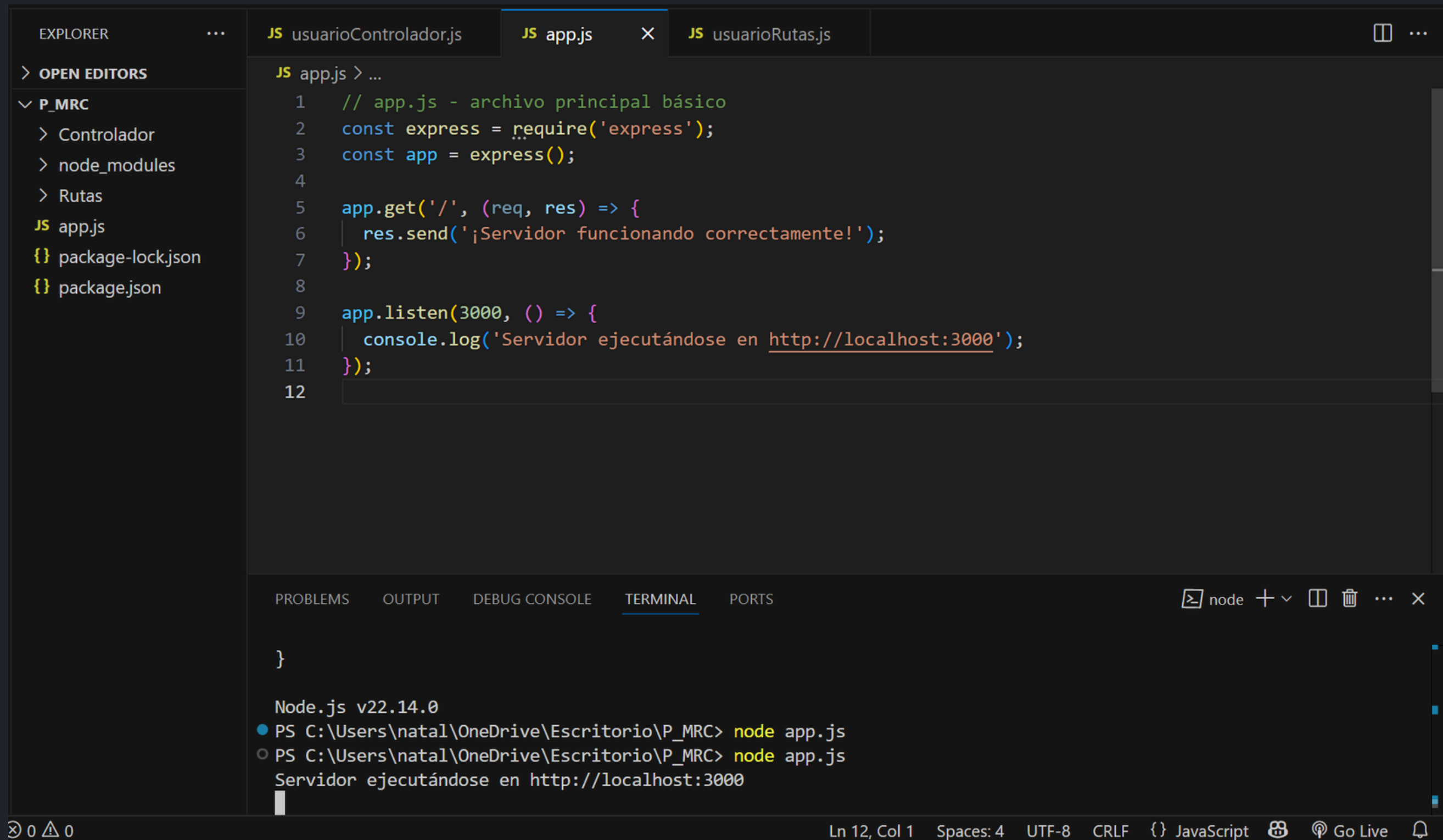
Dentro de P_MCR, se crean 3 carpetas app.js, controladores, rutas.

Dentro de cada una de las carpetas se crea un archivo con los siguientes nombres:
usuarioController.js
usuarioRoutes.js
app.js



Y dentro de cada archivo se agrega su código correspondiente

Por ultimo se ejecuta en la terminal el comando `node app.js`



The image shows a Visual Studio Code editor window. The Explorer sidebar on the left shows a project named 'P_MRC' with a file tree containing 'Controlador', 'node_modules', 'Rutas', 'app.js', 'package-lock.json', and 'package.json'. The main editor area has three tabs: 'usuarioControlador.js', 'app.js' (active), and 'usuarioRutas.js'. The 'app.js' file contains the following JavaScript code:

```
1 // app.js - archivo principal básico
2 const express = require('express');
3 const app = express();
4
5 app.get('/', (req, res) => {
6   res.send('¡Servidor funcionando correctamente!');
7 });
8
9 app.listen(3000, () => {
10   console.log('Servidor ejecutándose en http://localhost:3000');
11 });
12
```

Below the editor, the 'TERMINAL' tab is active, showing the command prompt output:

```
Node.js v22.14.0
PS C:\Users\natal\OneDrive\Escritorio\P_MRC> node app.js
PS C:\Users\natal\OneDrive\Escritorio\P_MRC> node app.js
Servidor ejecutándose en http://localhost:3000
```

The status bar at the bottom indicates the current line and column: 'Ln 12, Col 1'. Other status information includes 'Spaces: 4', 'UTF-8', 'CRLF', and 'JavaScript'.

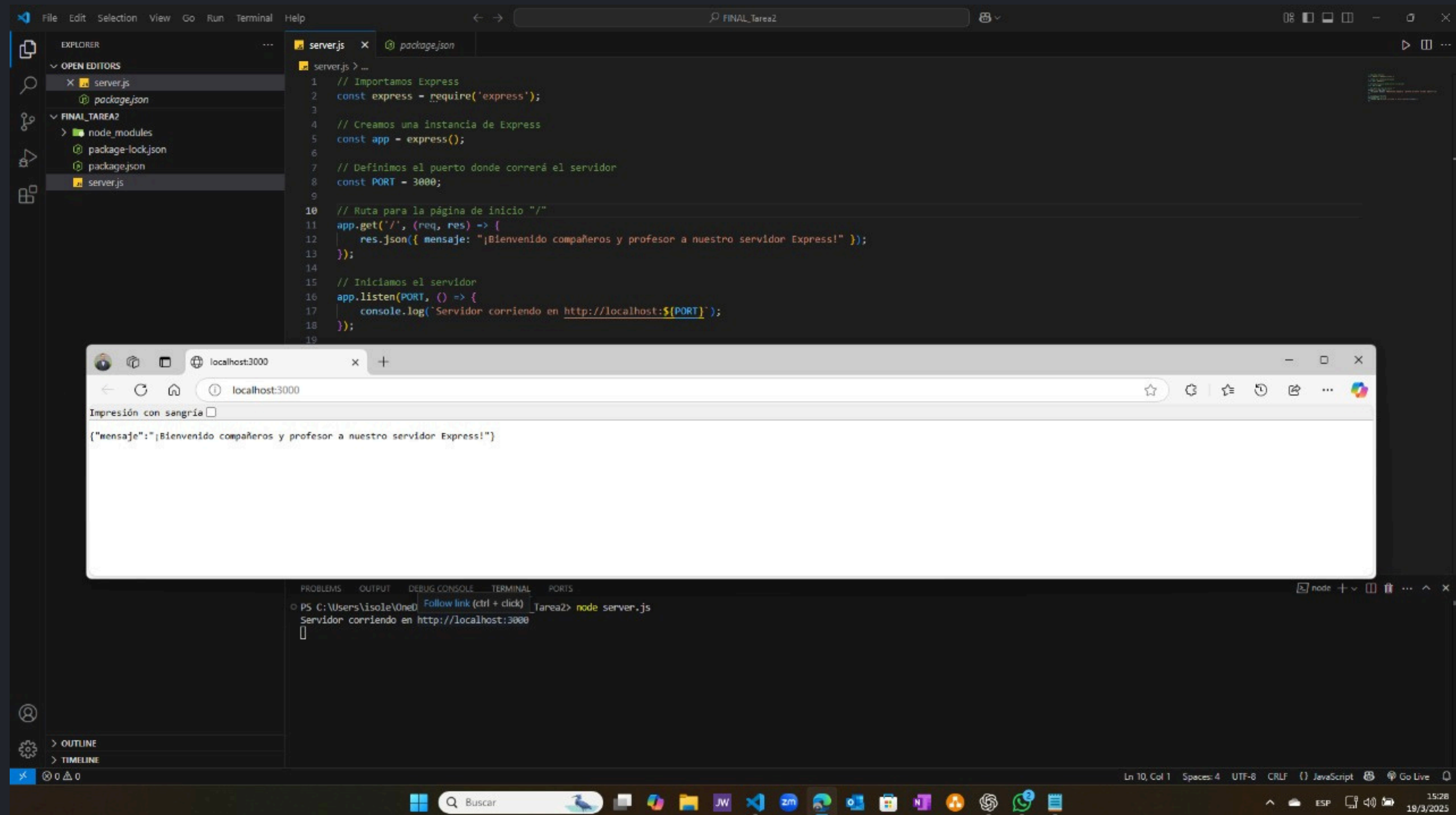
3.Diferencias entre una aplicación con Node.js puro y una con Express. {

Con Node.js puro se obtiene un control total sobre cada aspecto del servidor lo que hace que se tenga que escribir manualmente tareas comunes como el manejo de rutas, encabezados, o análisis de datos. Lo que resulta en invertir más tiempo y código

Express.js abstrae y simplifica tareas comunes y permite construir aplicaciones rápidamente, proporciona soporte para middlewares ofrece herramientas y patrones listos para usar por lo tanto requiere de menos código. Pero al abstraer muchas tareas puede sentirse limitado.

}

4. Demostración práctica: Crear un servidor Express con una ruta hola que devuelva un mensaje JSON.



```
<!--Programación IV-->
```

Muchas Gracias {

```
Grupo="#3"/>
```

}