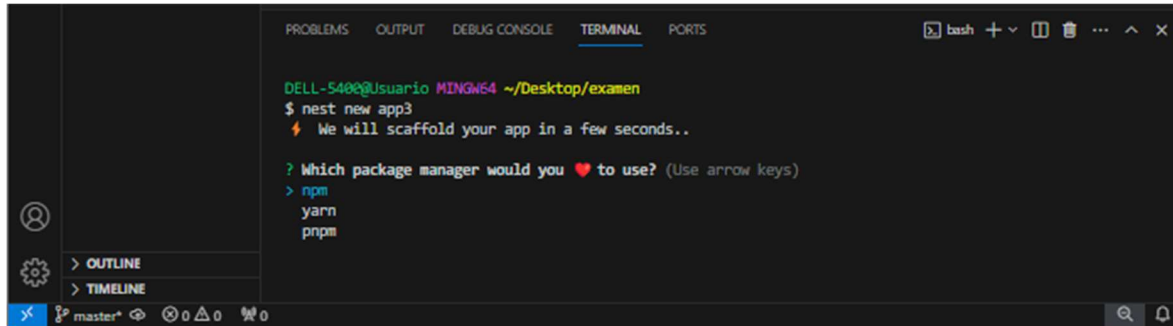


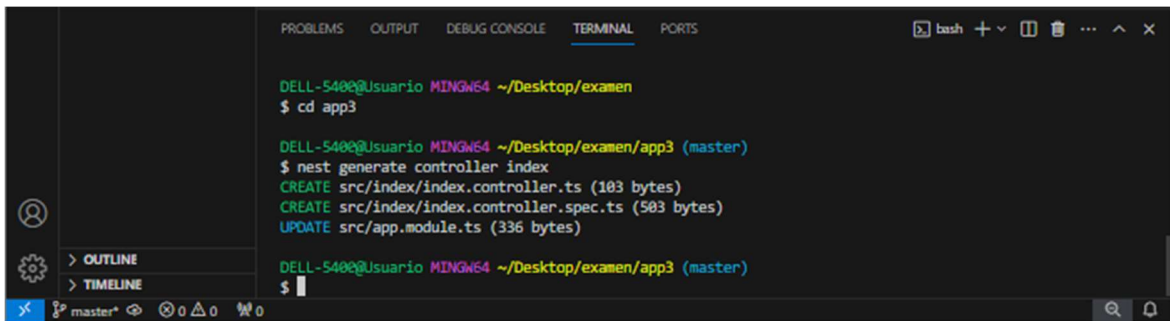
1.- Crea la aplicación en Nest y selecciona **npm**



```
DELL-5400@Usuario MINGW64 ~/Desktop/examen
$ nest new app3
⚡ We will scaffold your app in a few seconds..

? Which package manager would you ❤️ to use? (Use arrow keys)
> npm
  yarn
  pnpm
```

2.- métete a la carpeta de tu app con “cd [nombre\_app]” y crea tu controlador

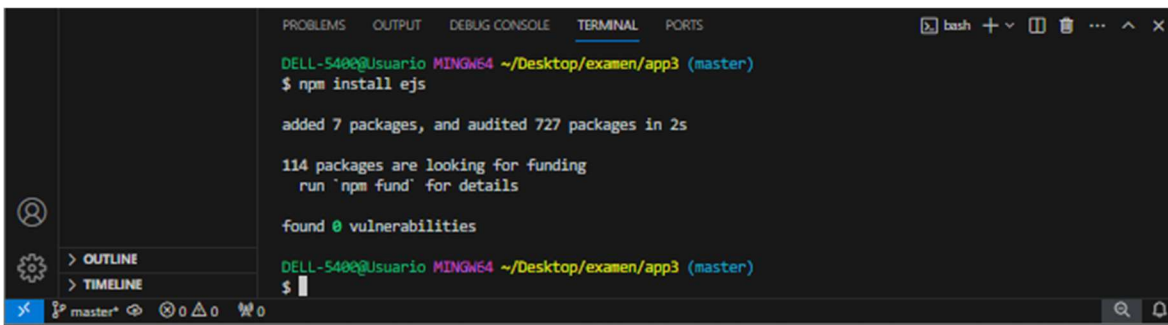


```
DELL-5400@Usuario MINGW64 ~/Desktop/examen
$ cd app3

DELL-5400@Usuario MINGW64 ~/Desktop/examen/app3 (master)
$ nest generate controller index
CREATE src/index/index.controller.ts (103 bytes)
CREATE src/index/index.controller.spec.ts (503 bytes)
UPDATE src/app.module.ts (336 bytes)

DELL-5400@Usuario MINGW64 ~/Desktop/examen/app3 (master)
$
```

3.- instala ejs



```
DELL-5400@Usuario MINGW64 ~/Desktop/examen/app3 (master)
$ npm install ejs

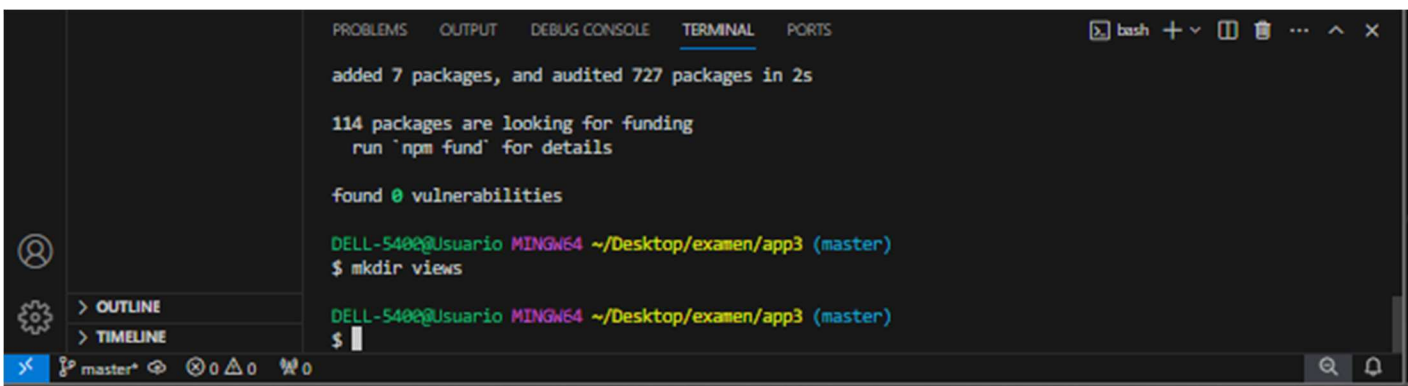
added 7 packages, and audited 727 packages in 2s

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

found 0 vulnerabilities

DELL-5400@Usuario MINGW64 ~/Desktop/examen/app3 (master)
$
```

4.- con **mkdir** crea la carpeta views



```
added 7 packages, and audited 727 packages in 2s

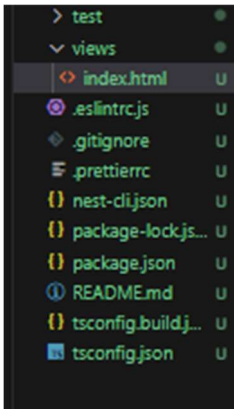
114 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

DELL-5400@Usuario MINGW64 ~/Desktop/examen/app3 (master)
$ mkdir views

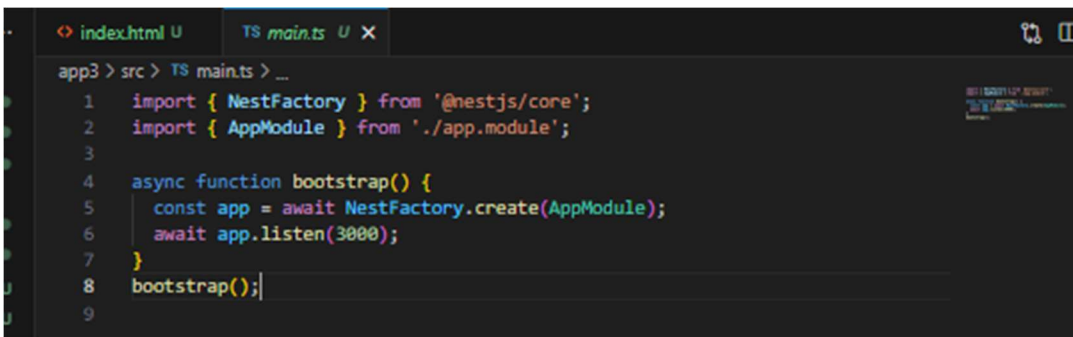
DELL-5400@Usuario MINGW64 ~/Desktop/examen/app3 (master)
$
```

5.- añade el html con el nombre del controlador, en este caso, index



6.- Abrirás **main.ts** que está en la carpeta **src** y configura el motor de plantillas para renderizar html

Pasando de esto:

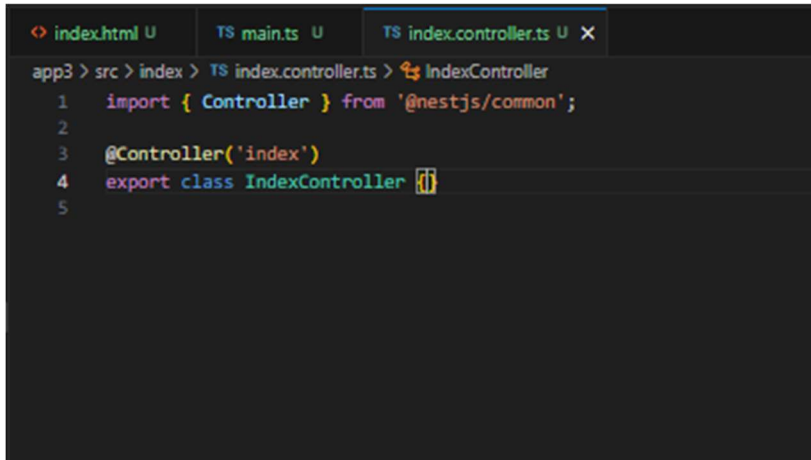


A esto:



7.- Ahora, definimos rutas en el controlador:

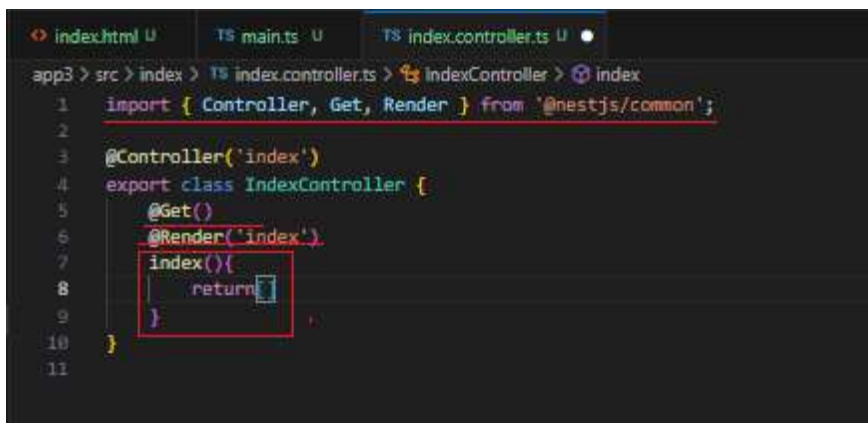
Pasando de esto:



The screenshot shows a VS Code editor with three tabs: 'index.html', 'TS main.ts', and 'TS index.controller.ts'. The active tab is 'TS index.controller.ts', which contains the following code:

```
app3 > src > index > TS index.controller.ts > IndexController
1  import { Controller } from '@nestjs/common';
2
3  @Controller('index')
4  export class IndexController {}
5
```

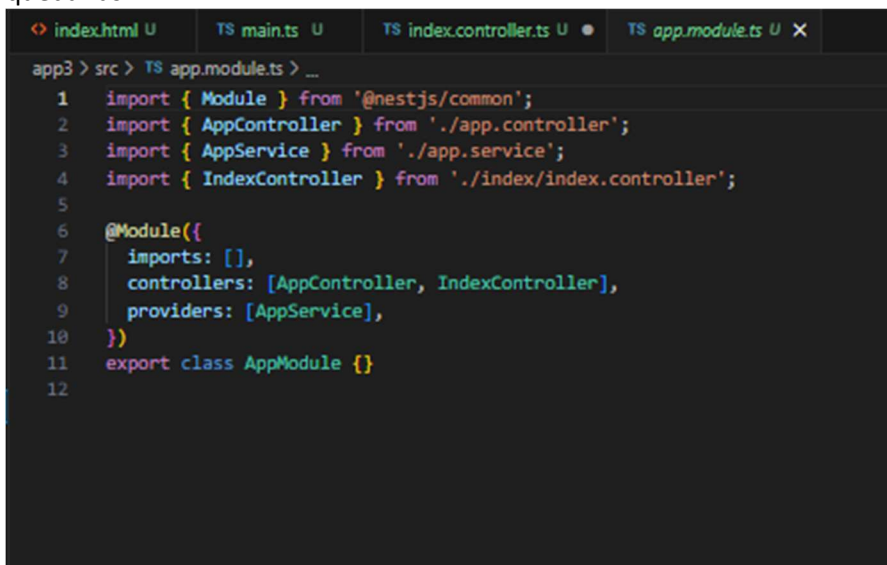
a esto:



The screenshot shows the same VS Code editor, but the 'TS index.controller.ts' tab now contains more code. The 'index' tab is also active in the breadcrumb. The code is as follows:

```
app3 > src > index > TS index.controller.ts > IndexController > index
1  import { Controller, Get, Render } from '@nestjs/common';
2
3  @Controller('index')
4  export class IndexController {
5      @Get()
6      @Render('index')
7      index(){
8          return;
9      }
10 }
11
```

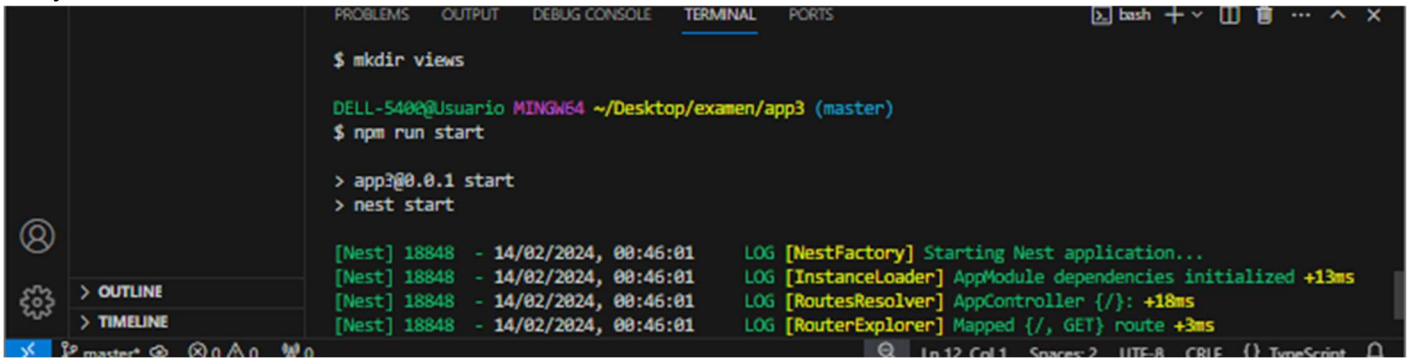
8.- Generalmente, al crear el controller, se registra automáticamente en el módulo principal, en caso que no, debe quedar así:



The screenshot shows a VS Code editor with four tabs: 'index.html', 'TS main.ts', 'TS index.controller.ts', and 'TS app.module.ts'. The active tab is 'TS app.module.ts', which contains the following code:

```
app3 > src > TS app.module.ts > ...
1  import { Module } from '@nestjs/common';
2  import { AppController } from '../app.controller';
3  import { AppService } from '../app.service';
4  import { IndexController } from '../index/index.controller';
5
6  @Module({
7      imports: [],
8      controllers: [AppController, IndexController],
9      providers: [AppService],
10 })
11 export class AppModule {}
12
```

9.- Ejecutamos:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
$ mkdir views
DELL-5400@Usuario MINGW64 ~/Desktop/examen/app3 (master)
$ npm run start

> app3@0.0.1 start
> nest start

[Nest] 18848 - 14/02/2024, 00:46:01 LOG [NestFactory] Starting Nest application...
[Nest] 18848 - 14/02/2024, 00:46:01 LOG [InstanceLoader] AppModule dependencies initialized +13ms
[Nest] 18848 - 14/02/2024, 00:46:01 LOG [RoutesResolver] AppController {/}: +18ms
[Nest] 18848 - 14/02/2024, 00:46:01 LOG [RouterExplorer] Mapped {/, GET} route +3ms
```

10.- e insertaremos el link (en mi caso) localhost:3000/index

