Universidad Autónoma de Chiapas
Facultad de contaduría y administración
campus I Tuxtla Gutiérrez

licenciatura en ingenieria en desarrollo y tecnologia de software

asignatura:

Taller de desarrollo 4

Actividad:

Documentacion

Nombre del catedratico:

D.s.c Luis Gutiérrez Alfaro

Nombre del alumno:

Garcia Muñoz Cristian Adayr

Numero de control:

A210637

Semestre y grupo: 6-No

Tuxtla Gutiérrez Chiapas

21/02/2024

Documentación.

Como primer paso lo que tenemos que hacer es abrir el power Shell en modo administrador de preferencia y ingresamos los siguientes comandos.

```
PS C:\Windows\system32> cd..
PS C:\Windows> cd..
PS C:\> ls
```

los comandos son para movernos entre directorios y ver las lista de los documentos con los que contamos.

```
    Directorio: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----     18/08/2023  06:22 p. m.                $WINDOWS.~BT
d-----     10/09/2023  07:49 a. m.                C
d-----     16/02/2024  06:00 p. m.                compiladoresFlask6n
d-----     22/11/2023  02:45 p. m.                Descargas
d-----     19/06/2023  01:04 a. m.                Drivers
d-----     29/08/2023  07:19 p. m.                emu8086
d-----     22/11/2023  02:44 p. m.                Escritorio
d-----     23/09/2023  07:37 p. m.                ESD
d-----     07/02/2024  04:48 p. m.                Intel
d-----     07/05/2022  12:24 a. m.                PerfLogs
d-r---     01/02/2024  05:43 p. m.                Program Files
d-r---     14/11/2023  11:13 p. m.                Program Files (x86)
d-----     08/11/2023  11:43 a. m.                respaldo
d-----     07/09/2023  04:20 p. m.                SQL2022
d-----     14/02/2024  01:19 p. m.                taller
```

```
PS C:\> cd ./taller/
PS C:\taller> npm i -g @nestjs/cli
```

el cd es para movernos dentro de la carpeta de taller, y el siguiente comando es para crear la estructura o instalar los paquetes, si se hace de manera correcta nos debe de aparecer lo siguiente..

```
changed 288 packages in 32s

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

Como siguiente paso creamos el nuevo proyecto.

```
PS C:\taller> nest new Examen
⚡  We will scaffold your app in a few seconds..
```

si nos sale bien nos debe salir lo siguiente.

```
? Which package manager would you ❤  to use? npm
CREATE examen/.eslintrc.js (688 bytes)
CREATE examen/.prettierrc (54 bytes)
CREATE examen/nest-cli.json (179 bytes)
CREATE examen/package.json (2014 bytes)
CREATE examen/README.md (3413 bytes)
CREATE examen/tsconfig.build.json (101 bytes)
CREATE examen/tsconfig.json (567 bytes)
CREATE examen/src/app.controller.ts (286 bytes)
CREATE examen/src/app.module.ts (259 bytes)
CREATE examen/src/app.service.ts (150 bytes)
CREATE examen/src/main.ts (216 bytes)
CREATE examen/src/app.controller.spec.ts (639 bytes)
CREATE examen/test/jest-e2e.json (192 bytes)
CREATE examen/test/app.e2e-spec.ts (654 bytes)

√ Installation in progress... ☕

🚀  Successfully created project examen
👉  Get started with the following commands:
```

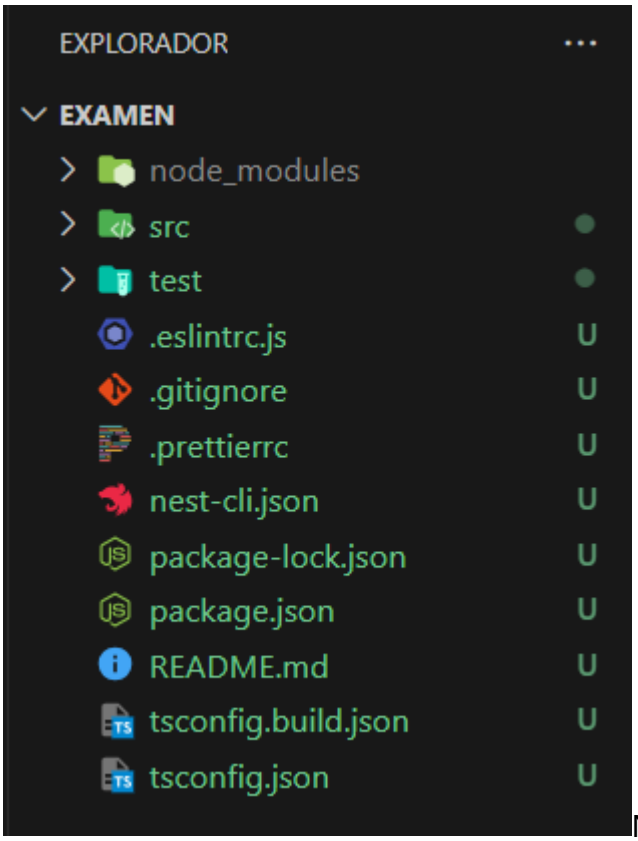
```
$ cd examen
$ npm run start


                        Thanks for installing Nest 🙏
                 Please consider donating to our open collective
                        to help us maintain this package.


               🍷  Donate: https://opencollective.com/nest
```

```
PS C:\taller> cd ./examen/
PS C:\taller\examen> code .
PS C:\taller\examen>
```

luego entramos al directorio de nuestro proyecto y lo abrimos con code . para empezar a modificar en el código.



Nos debería quedar de esta manera

A continuación crearemos, el controlador y el modulo con el comando nest g _ _

```
PS C:\taller\examen> nest g co
? What name would you like to use for the controller? E_Controlador
CREATE src/e_controlador/e_controlador.controller.ts (118 bytes)
CREATE src/e_controlador/e_controlador.controller.spec.ts (553 bytes)
UPDATE src/app.module.ts (366 bytes)
PS C:\taller\examen> nest g mo E_Modulo
CREATE src/e_modulo/e_modulo.module.ts (88 bytes)
UPDATE src/app.module.ts (439 bytes)
PS C:\taller\examen>
```

Ahora para crear una aplicación MVC, también necesitamos un motor de plantillas para representar nuestras vistas HTML asi que ejecutamos el siguiente comando.

```
PS C:\taller\examen> npm install --save hbs

added 7 packages, and audited 730 packages in 3s

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

found 0 vulnerabilities
```

Una vez que se completa el proceso de instalación, necesitamos configurar la instancia expresa usando el siguiente código:

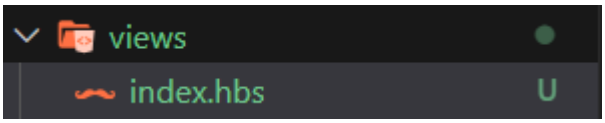
```
import { NestFactory } from '@nestjs/core';
import { NestExpressApplication } from '@nestjs/platform-express';
import { join } from 'path';
import { AppModule } from './app.module';

async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(
    AppModule,
  );

  app.useStaticAssets(join(__dirname, '..', 'public'));
  app.setBaseViewsDir(join(__dirname, '..', 'views'));
  app.setViewEngine('hbs');

  await app.listen(3000);
}
bootstrap();
```

Le dijimos a Express que el directorio público se usará para almacenar activos estáticos, las vistas contendrán plantillas y el motor de plantillas hbs debería usarse para representar la salida HTML. Luego lo que hacemos es crear nuestra carpeta views con un idenx.hbs dentro.



A continuación, abra el archivo app.controller y reemplace el método root() con el siguiente código.

```
import { Get, Controller, Render } from '@nestjs/common';

@Controller('index')
export class AppController {
  @Get()
  @Render('index')
  root() {
    return { message: 'Cristian Adayr Garcia Muñoz' };
  }
}
```

A continuación modificamos el app.module, de primeras los tendremos de las siguiente manera.

```
import { Module } from '@nestjs/common';
import { AppController } from './app.controller';
import { AppService } from './app.service';
import { EControladorController } from './e_controlador/e_controlador.controller';
import { EModuloModule } from './e_modulo/e_modulo.module';
import { ExaController } from './exa/exa.controller';
import { EMModule } from './e_m/e_m.module';

@Module({
  imports: [EModuloModule, EMModule],
  controllers: [AppController, EControladorController, ExaController],
  providers: [AppService],
})
export class AppModule {}

```

y lo deberemos de dejar de la siguiente manera.

```
import { Module } from '@nestjs/common';
import { AppController } from './app.controller';
//import { AppService } from './app.service';
//import { EControladorController } from './e_controlador/e_controlador.controller';
//import { EModuloModule } from './e_modulo/e_modulo.module';
import { ExaController } from './exa/exa.controller';
import { EMModule } from './e_m/e_m.module';

@Module({
  imports: [],
  controllers: [AppController],
  //providers: [AppService],
})
export class AppModule {}
```

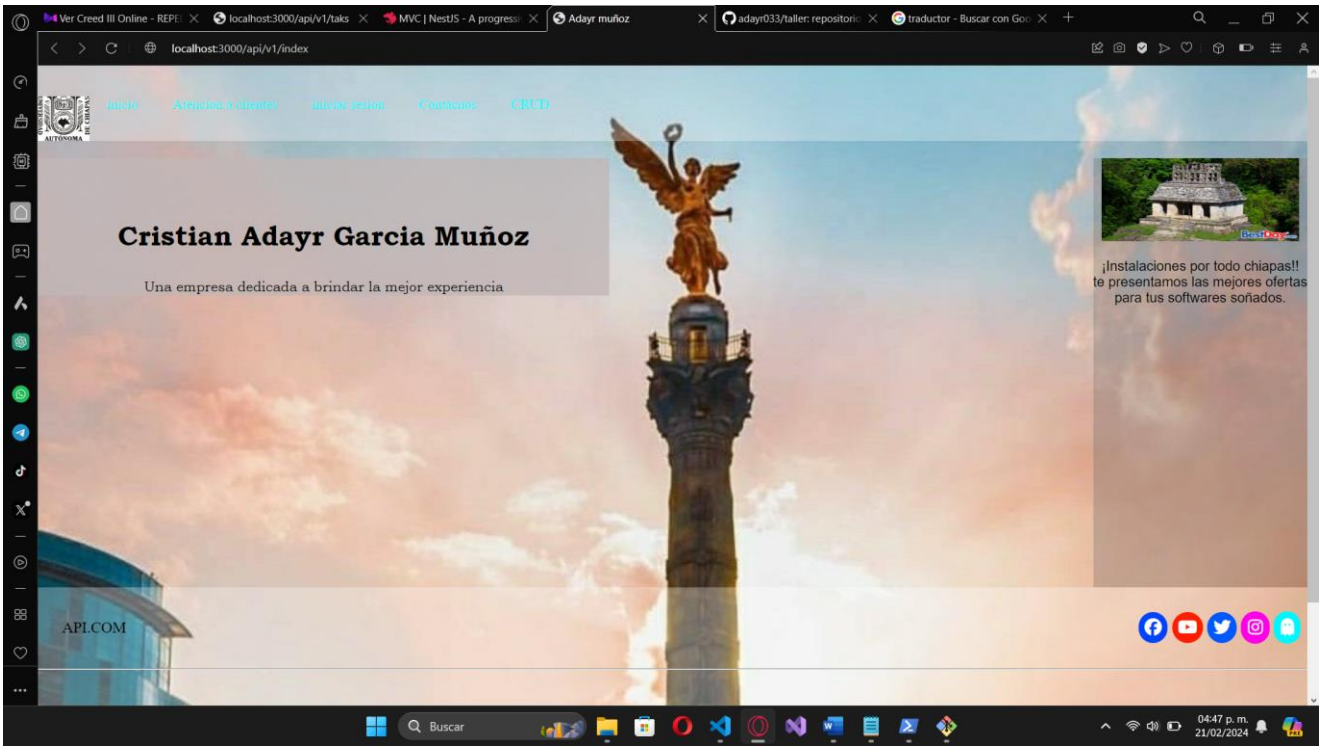
Comente lo que no es necesario tener y en la parte de module se elimina lo que este alado de appcontroller.

Ya para finalizar en el controlador procedemos a hacer nuestra ruta la mia quedo de la siguiente manera.

```
@Controller('api/v1/index')
```

npm run start:dev --- ejecutamos el proyecto con el siguiente comando que nos permite visualizar los cambios que hagamos sin necesidad de detener el proyecto.

Y nuestro proyecto nos debería quedar de la siguiente manera.


Adayr muñoz
localhost:3000/api/v1/index
inicio
Atencion a clientes
iniciar sesion
Contactos
CRUD
Cristian Adayr Garcia Muñoz
Una empresa dedicada a brindar la mejor experiencia
¡Instalaciones por todo chiapas!!
te presentamos las mejores ofertas
para tus softwares soñados.
APLCOM
Buscar
04:47 p. m.
21/02/2024