



Universidad Autónoma de Chiapas

Licenciatura en Ingeniería en Desarrollo y Tecnología de Software, Campus 1



Tuxtla Gutiérrez Chiapas, 18 de Abril del 2024

Nombre del Docente:

GUTIERREZ ALFARO LUIS, DR

Nombre de la Materia:

Taller de Desarrollo 4

Nombre del Alumno:

Clemente López Jasson Jared

Semestre y Grupo:

6-M

Actividad 3.1 MicroServicio con BD

(link de github)

https://github.com/Jasson2003/Actividad-3_1-MicroServicio-con-BD.git

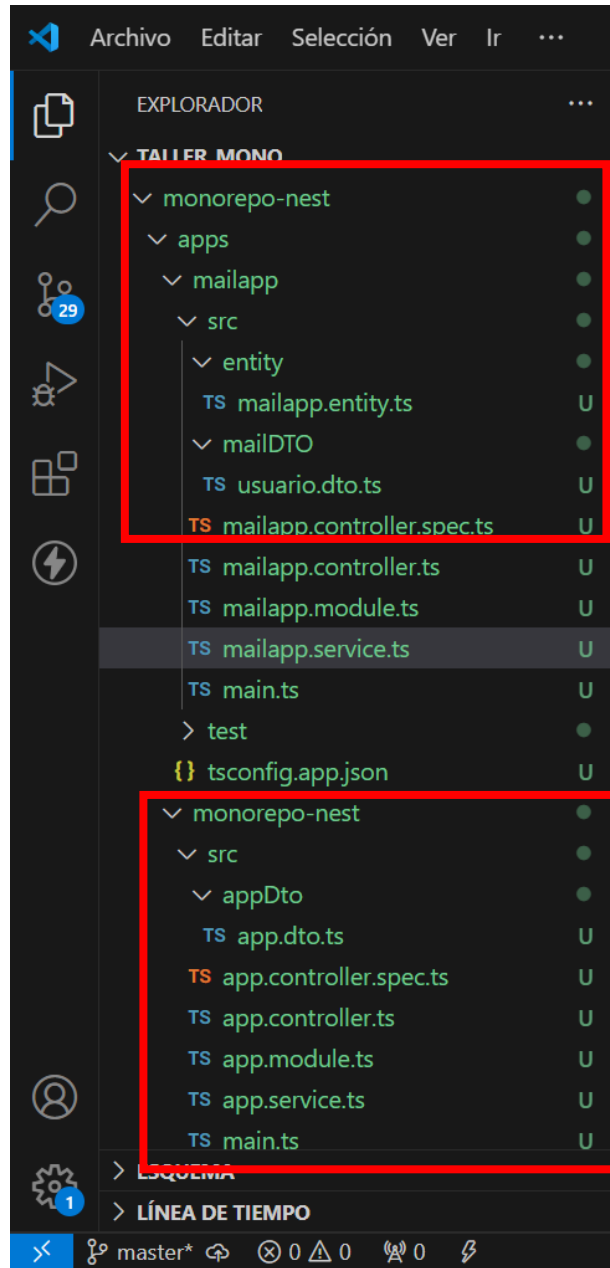
Dentro de mi archivo instale:

```
PS C:\Users\jasso\OneDrive\Documentos\MicroBD\microbd> npm install --save @nestjs/typeorm mysql2 typeorm

added 34 packages, and audited 749 packages in 19s

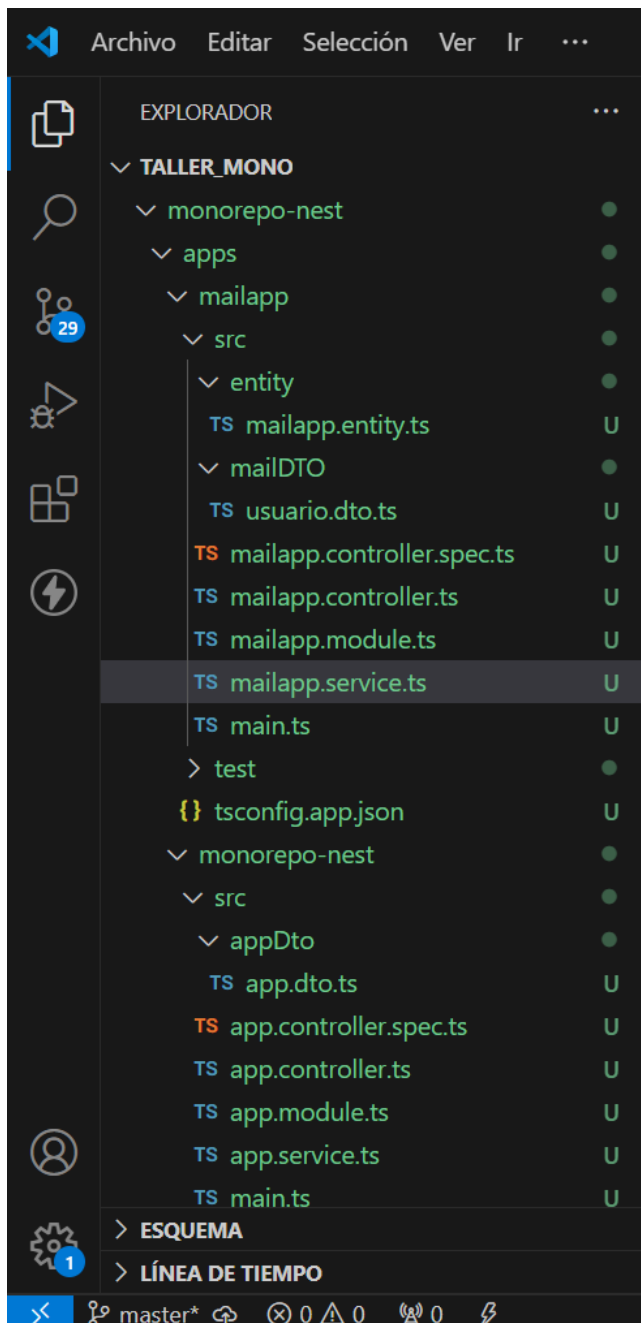
121 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\jasso\OneDrive\Documentos\MicroBD\microbd> yarn install
yarn install v1.22.19
info No lockfile found.
```

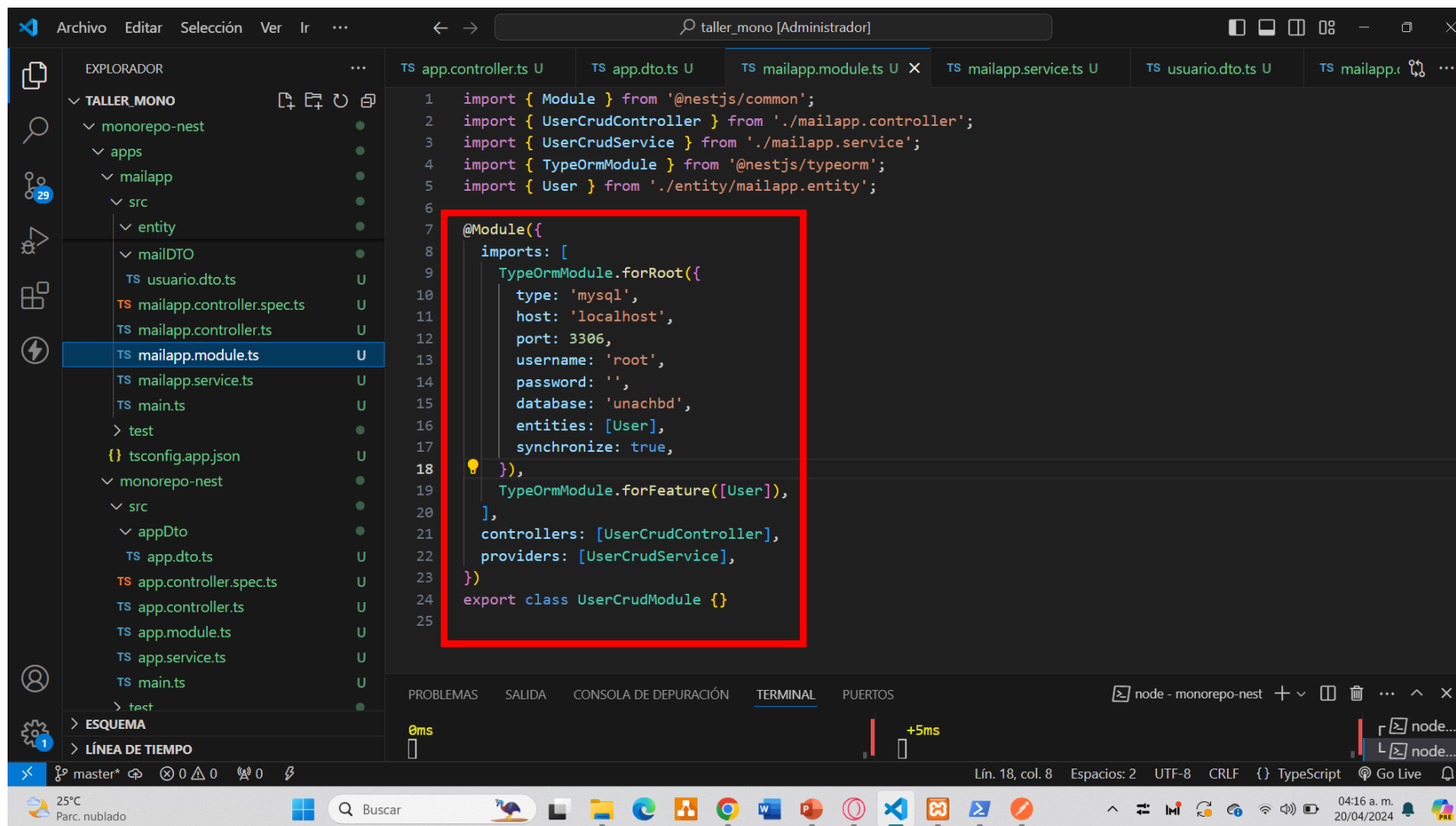


Secundario

PRINCIPAL

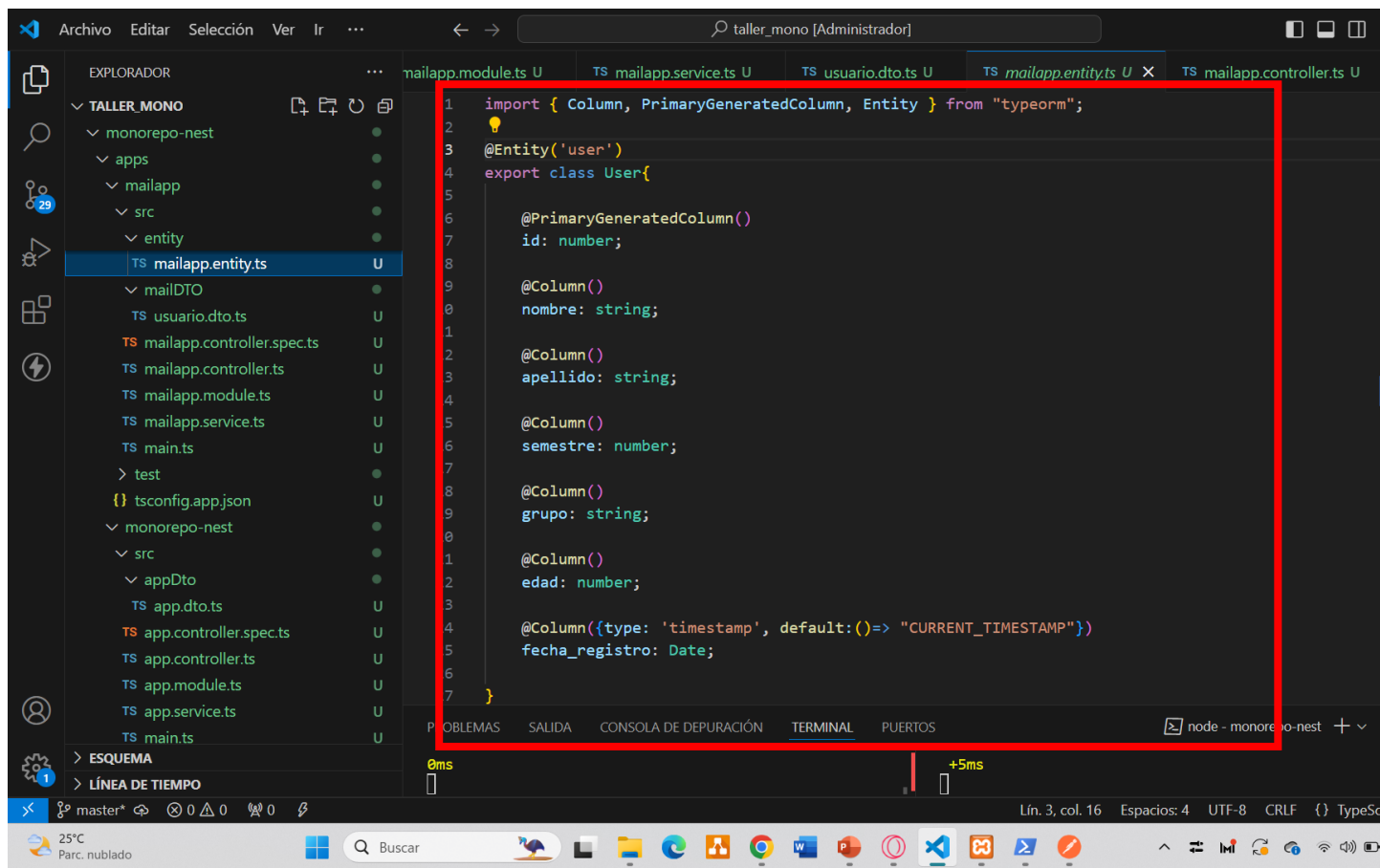


El código de cada archivo se subió al Github, pero en conclusión: se hizo una conexión a mysql y se utilizo de igual manera TCP para la creación de este microservicio.



```
1 import { Module } from '@nestjs/common';
2 import { UserCrudController } from '../mailapp.controller';
3 import { UserCrudService } from '../mailapp.service';
4 import { TypeOrmModule } from '@nestjs/typeorm';
5 import { User } from '../entity/mailapp.entity';
6
7 @Module({
8   imports: [
9     TypeOrmModule.forRoot({
10       type: 'mysql',
11       host: 'localhost',
12       port: 3306,
13       username: 'root',
14       password: '',
15       database: 'unachbd',
16       entities: [User],
17       synchronize: true,
18     }),
19     TypeOrmModule.forFeature([User]),
20   ],
21   controllers: [UserCrudController],
22   providers: [UserCrudService],
23 })
24 export class UserCrudModule {}
25
```

este código establece una arquitectura básica para la gestión de usuarios en una aplicación NestJS, proporcionando una estructura clara y organizada para las operaciones CRUD en la base de datos MySQL



```
1 import { Column, PrimaryGeneratedColumn, Entity } from "typeorm";
2
3 @Entity('user')
4 export class User{
5
6     @PrimaryGeneratedColumn()
7     id: number;
8
9     @Column()
10    nombre: string;
11
12    @Column()
13    apellido: string;
14
15    @Column()
16    semestre: number;
17
18    @Column()
19    grupo: string;
20
21    @Column()
22    edad: number;
23
24    @Column({type: 'timestamp', default: () => "CURRENT_TIMESTAMP"})
25    fecha_registro: Date;
26
27 }
```

Esta clase define la estructura de la tabla “user”, especificando los tipos de datos y las propiedades de cada columna, utilizando el ORM TypeORM en una aplicación Node.js.

```
1 import { Controller, Get, Inject, Post, Patch, Delete, Body, Put } from '@nestjs/common';
2 import { AppService } from './app.service';
3 import { UserDto } from './appDto/app.dto';
4 import { ClientProxy } from '@nestjs/microservices';
5
6 @Controller()
7 export class AppController {
8   constructor(
9     private readonly appService: AppService,
10    @Inject('USERCRUD') private userCrud: ClientProxy
11   ) {}
12
13   @Post('create')
14   async create(@Body() userDto : UserDto){
15     console.log(userDto);
16     const servieName = {cdm: 'create'};
17     var result = await this.userCrud.send(servieName, userDto);
18     return result;
19   }
20
21   @Get('findOne')
22   async findOne(@Body() id : number){
23     const servieName = {cdm: 'findOne'};
24     var result = await this.userCrud.send(servieName, id);
25     return result;
26   }
27 }
```

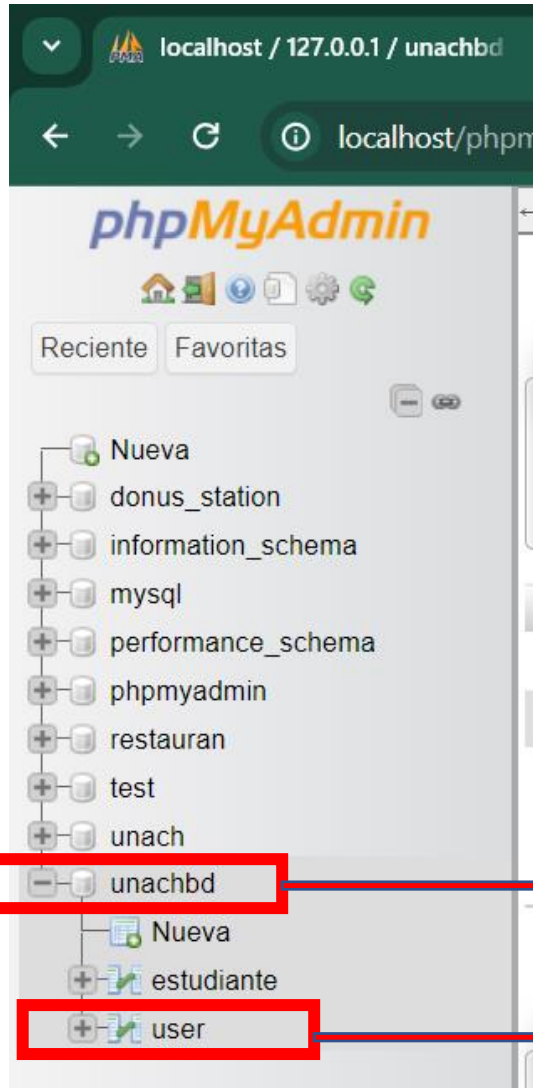
Este controlador proporciona una interfaz de API para interactuar con un servicio de gestión de usuarios a través de microservicios, permitiendo crear, leer, actualizar y eliminar usuarios utilizando HTTP.

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORADOR (File Explorer):** Displays the project structure under 'TALLER_MONO'. The 'src' folder is expanded, showing files like 'entity', 'mailapp.entity.ts', 'mailDTO', 'usuario.dto.ts', 'mailapp.controller.spec.ts', 'mailapp.controller.ts', 'mailapp.module.ts', 'mailapp.service.ts', and 'main.ts'.
- Source Code:** The 'main.ts' file is open, showing imports for '@nestjs/common', '@nestjs/typeorm', 'typeorm', 'Repository', 'UserDto', and 'User'.
- TERMINAL:** The terminal output shows the command 'nest start monorepo-nest' and the subsequent compilation and startup of the application. The output includes logs for 'webpack 5.90.1 compiled successfully', 'NestFactory Starting Nest application...', 'ClientsModule dependencies initialized', 'AppModule dependencies initialized', 'AppController {/}: +7ms', 'Mapped {/create, POST} route +3ms', 'Mapped {/findOne, GET} route +0ms', 'Mapped {/update, PUT} route +1ms', 'Mapped {/remove, DELETE} route +0ms', and 'Nest application successfully started +3ms'.

Levantamos los servicios
(Primario y Secundario)

AHORA EN POSTMAN



id	nombre	semestre	grupo	edad	fecha_registro	apellido
----	--------	----------	-------	------	----------------	----------

Las columnas que se crearon en .entity.ts

GET



http://localhost:3001/findOne

Send



```
1 {  
2  
3   "id": "4"  
4 }  
5
```

Consultar Registro con
el id

Body Cookies Headers (7) Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "id": 4,  
3   "nombre": "Jasson Jared ",  
4   "apellido": "Clemente Lopez",  
5   "semestre": 6,  
6   "grupo": "m",  
7   "edad": 30,  
8   "fecha_registro": "2024-04-20T10:51:42.000Z"  
9 }
```

POST

▼

http://localhost:3001/create

Send

▼

Con el Post creamos nuevos registros y cuando actualizamos la BD nos aparece la columna que agregamos.

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "nombre": "Jasson",
3   "apellido": "Clemente",
4   "semestre": 6,
5   "grupo": "m",
6   "edad": 20
7 }
```

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "nombre": "Jasson ",
3   "apellido": "Clemente",
4   "semestre": 6,
5   "grupo": "m",
6   "edad": 20,
7   "id": 4,
8   "fecha_registro": "2024-04-20T10:51:42.000Z"
9 }
```

Nueva

donus_station

information_schema

mysql

performance_schema

phpmyadmin

restauran

test

unach

unachbd

Nueva

estudiante

user

SELECT * FROM `user`

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo

Número de filas: 25

Filtrar filas: Buscar en esta tabla

Opciones extra

	id	nombre	semestre	grupo	edad	fecha_registro	apellido
<input type="checkbox"/>	4	Jasson	6	m	20	2024-04-20 04:51:42	Clemente

Seleccionar todo

Para los elementos que están marcados:

Editar

Copiar

Borrar

Exportar

Mostrar todo

Número de filas: 25

Filtrar filas: Buscar en esta tabla

Operaciones sobre los resultados de la consulta

Imprimir

Copiar al portapapeles

Exportar

Mostrar gráfico

Crear vista

```
6
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
node - monorepo-nest + v
Enfocar carpeta en el explorador (ctrl + clic)
PS C:\Users\jasso\OneDrive\Documentos\taller_mono\monorepo-nest> nest start mailapp
webpack 5.90.1 compiled successfully in 7479 ms
[Nest] 17940 - 20/04/2024, 4:42:58 a.m. LOG [NestFactory] Starting Nest application...
[Nest] 17940 - 20/04/2024, 4:42:58 a.m. LOG [InstanceLoader] TypeOrmModule dependencies initialized +238ms
[Nest] 17940 - 20/04/2024, 4:42:59 a.m. LOG [InstanceLoader] TypeOrmCoreModule dependencies initialized +636ms
[Nest] 17940 - 20/04/2024, 4:42:59 a.m. LOG [InstanceLoader] TypeOrmModule dependencies initialized +0ms
[Nest] 17940 - 20/04/2024, 4:42:59 a.m. LOG [InstanceLoader] UserCrudModule dependencies initialized +1ms
[Nest] 17940 - 20/04/2024, 4:42:59 a.m. LOG [NestMicroservice] Nest microservice successfully started +15ms
[
  {
    nombre: 'Jasson ',
    apellido: 'Clemente',
    semestre: 6,
    grupo: 'm',
    edad: 20
  },
  {
    nombre: 'Alejandro',
    apellido: 'Cruz',
    semestre: 9,
    grupo: 'H',
    edad: 25
  },
  {
    nombre: 'Fatima',
    apellido: 'Escobar',
    semestre: 5,
    grupo: 'N',
    edad: 22
  }
]
Lín. 32, col. 1 Espacios: 2 UTF-8 CRLF {} TypeScript
```

Para seguir mostrando se agregaron mas registros.

<div><div></div><div></div><div></div></div>				id	nombre	semestre	grupo	edad	fecha_registro	apellido			
<input type="checkbox"/>		Editar		Copiar		Borrar	4	Jasson	6	m	20	2024-04-20 04:51:42	Clemente
<input type="checkbox"/>		Editar		Copiar		Borrar	5	Alejandro	9	H	25	2024-04-20 04:57:11	Cruz
<input type="checkbox"/>		Editar		Copiar		Borrar	6	Fatima	5	N	22	2024-04-20 04:57:57	Escobar

DELETE

▼

http://localhost:3001/remove

Send

▼

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

4

5

6

{

....

...."id": "5"

}

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

▼

⌵

1

2

3

4

{

"raw": [],

"affected": 1







}

Con delete eliminamos registro solo poniendo en raw el id del registro por borrar, como nos podemos dar cuenta el antes y después en la tabla.

ANTES

<div><div><div>←T→</div><div>▼</div></div><div><div>id</div><div>nombre</div><div>semestre</div><div>grupo</div><div>edad</div><div>fecha_registro</div><div>apellido</div></div></div>										
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	4	Jasson	6	m	20	2024-04-20 04:51:42	Clemente
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5	Alejando	9	H	25	2024-04-20 04:57:11	Cruz
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	6	Fatima	5	N	22	2024-04-20 04:57:57	Escobar

DESPUES

<div><div>←T→</div><div></div></div>				id	nombre	semestre	grupo	edad	fecha_registro	apellido
<input type="checkbox"/>				4	Jasson		6 m	20	2024-04-20 04:51:42	Clemente
<input type="checkbox"/>				6	Fatima		5 N	22	2024-04-20 04:57:57	Escobar

PUT



http://localhost:3001/update

Send



```
1 {
2   ...."id": "4",
3   ...."nombre": "Jasson Jared ",
4   ...."apellido": "Clemente Lopez",
5   ...."semestre": 6,
6   ...."grupo": "m",
7   ...."edad": 30
8
9
10 }
```

Body Cookies Headers (7) Test Results

Pretty

Raw

Preview

Visualize



JSON



```
1 {
2   "id": 4,
3   "nombre": "Jasson Jared ",
4   "apellido": "Clemente Lopez",
5   "semestre": 6,
6   "grupo": "m",
7   "edad": 30,
8   "fecha_registro": "2024-04-20T10:51:42.000Z"
9 }
```

Aquí podemos editar algún registro.

ANTES

			id	nombre	semestre	grupo	edad	fecha_registro	apellido	
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	4	Jasson	6	m	20	2024-04-20 04:51:42	Clemente
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	6	Fatima	5	N	22	2024-04-20 04:57:57	Escobar

DESPUES

						id	nombre	apellido	semestre	grupo	edad	fecha_registro	
<input type="checkbox"/>		Editar		Copiar		Borrar	4	Jasson Jared	Clemente Lopez	6	m	30	2024-04-20 04:51:42
<input type="checkbox"/>		Editar		Copiar		Borrar	6	Fatima	Escobar	5	N	22	2024-04-20 04:57:57