



UNIVERSIDAD AUTÓNOMA DE CHIAPAS.



FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN, CAMPUS I.

**LICENCIATURA EN INGENIERÍA EN DESARROLLO Y TECNOLOGÍAS
DE SOFTWARE.**

SEXTO SEMESTRE, GRUPO: "M"

MATERIA: TALLER DE DESARROLLO 4.

DOCENTE: DR. LUIS GUTIÉRREZ ALFARO.

ALUMNA: CIGARROA HERNÁNDEZ LUISA FERNANDA (A210118).

"EXAMEN TALLERES, MICROSERVICIO CON BD"

FECHA DE ENTREGA: VIERNES 19 DE ABRIL DE 2024.

CAPTURAS DEL CÓDIGO

SE MODIFICO EL APP.CONTROLLER.TS

```
apps > monorepo_project > src > app.controller.ts > AppController > newUser
1  import { Body, Controller, Post } from '@nestjs/common';
2  import { AppService } from './app.service';
3
4  @Controller() // Marca esta clase como un controlador
5  export class AppController {
6      constructor(private readonly _appService: AppService) {
7          // Constructor para inyectar AppService
8      }
9
10     @Post('new-user') // Define un endpoint POST accesible en /new-user
11     public async newUser(@Body() body: unknown) {
12         // Objeto de usuario falso predeterminado
13         const fakeUser = {
14             email: 'fake_user@microservice.com',
15             name: 'Fake User',
16             avatar: 'https://fake-image.com',
17             password: 'FakePassword_1234'
18         };
19
20         // Si el cuerpo de la solicitud está vacío, asigna fakeUser a body
21         if (!body || Object.keys(body).length === 0) {
```

```
app.controller.ts x app.service.ts app.module.ts message_app.service.ts app.controll
apps > monorepo_project > src > app.controller.ts > AppController > newUser
20     // Si el cuerpo de la solicitud está vacío, asigna fakeUser a body
21     if (!body || Object.keys(body).length === 0) {
22         body = fakeUser;
23     }
24
25     // Verifica que body es un objeto con las propiedades name y email
26     if (typeof body === 'object' && body !== null) {
27         const { name, email } = body as { name: string; email: string };
28
29         if (typeof name === 'string' && typeof email === 'string') {
30             // Llama al método newUser de AppService con el cuerpo modificado
31             return this._appService.nuevoUsauario({ name, email });
32         }
33     }
34
35     // Lanza un error si body no tiene el formato esperado
36     throw new Error('El formato de body es inválido o incompleto.');
```

SE MODIFICO EL APP.MODULE.TS

```
apps > monorepo_project > src > app.module.ts > AppModule
1  import { Module } from '@nestjs/common';
2  import { ClientsModule, Transport } from '@nestjs/microservices';
3  import { AppController } from './app.controller';
4  import { AppService } from './app.service';
5  import { TypeOrmModule } from '@nestjs/typeorm';
6  import { User } from './user.entity'; // Importa la entidad User
7
8  @Module({
9    imports: [
10     // Configura la conexión a la base de datos MySQL
11     TypeOrmModule.forRoot({
12       type: 'mysql',
13       host: 'localhost',
14       port: 3306,
15       username: 'root',
16       database: 'Examen_Taller',
17       entities: [User],
18       synchronize: true,
19     }),
20     // Configura ClientsModule y registra MESSAGE_SERVICE
21     ClientsModule.register([
```

```
apps > monorepo_project > src > app.module.ts > AppModule
20 // Configura ClientsModule y registra MESSAGE_SERVICE
21 ClientsModule.register([
22   {
23     name: 'MESSAGE_SERVICE',
24     transport: Transport.TCP,
25     options: {
26       host: 'localhost', // Asegúrate de usar la dirección correcta para el servicio
27       port: 3001, // Asegúrate de usar el puerto correcto para el servicio
28     },
29   },
30 ],
31 TypeOrmModule.forFeature([User]), // Importa tu entidad User
32 ],
33 controllers: [AppController],
34 providers: [AppService],
35 })
36 export class AppModule {}
37
```

SE MODIFICO APP.SERVICE.TS

```
app.controller.ts  app.service.ts x  app.module.ts  message_app.service.ts  app.controller.s
apps > monorepo_project > src > app.service.ts > AppService > nuevoUsuario
1  import { Inject, Injectable } from '@nestjs/common';
2  import { ClientProxy } from '@nestjs/microservices';
3  import { InjectRepository } from '@nestjs/typeorm';
4  import { Repository } from 'typeorm';
5  import { User } from './user.entity';
6
7  @Injectable()
8  export class AppService {
9    constructor(
10     @Inject('MESSAGE_SERVICE') private readonly _mailClient: ClientProxy,
11     @InjectRepository(User) private readonly userRepository: Repository<User>,
12   ) {}
13
14   async nuevoUsuario(body: { name: string; email: string }) {
15     // Inserta el usuario en la base de datos
16     const newUser = this.userRepository.create(body);
17     const savedUser = await this.userRepository.save(newUser);
18
19     // Emite el evento "new_email" al servicio
20     this._mailClient.emit('new_email', savedUser);
21
```

```
app.controller.ts  app.service.ts x  app.module.ts  message_app.service.ts  app.controller.spec.ts
apps > monorepo_project > src > app.service.ts > AppService > nuevoUsuario
8  export class AppService {
9    constructor(
10     @Inject('MESSAGE_SERVICE') private readonly _mailClient: ClientProxy,
11     @InjectRepository(User) private readonly userRepository: Repository<User>,
12   ) {}
13
14   async nuevoUsuario(body: { name: string; email: string }) {
15     // Inserta el usuario en la base de datos
16     const newUser = this.userRepository.create(body);
17     const savedUser = await this.userRepository.save(newUser);
18
19     // Emite el evento "new_email" al servicio
20     this._mailClient.emit('new_email', savedUser);
21
22     return savedUser;
23   }
24 }
25
```

COMPROBAMOS QUE ESTE CORRIENDO EL PROGRAMA

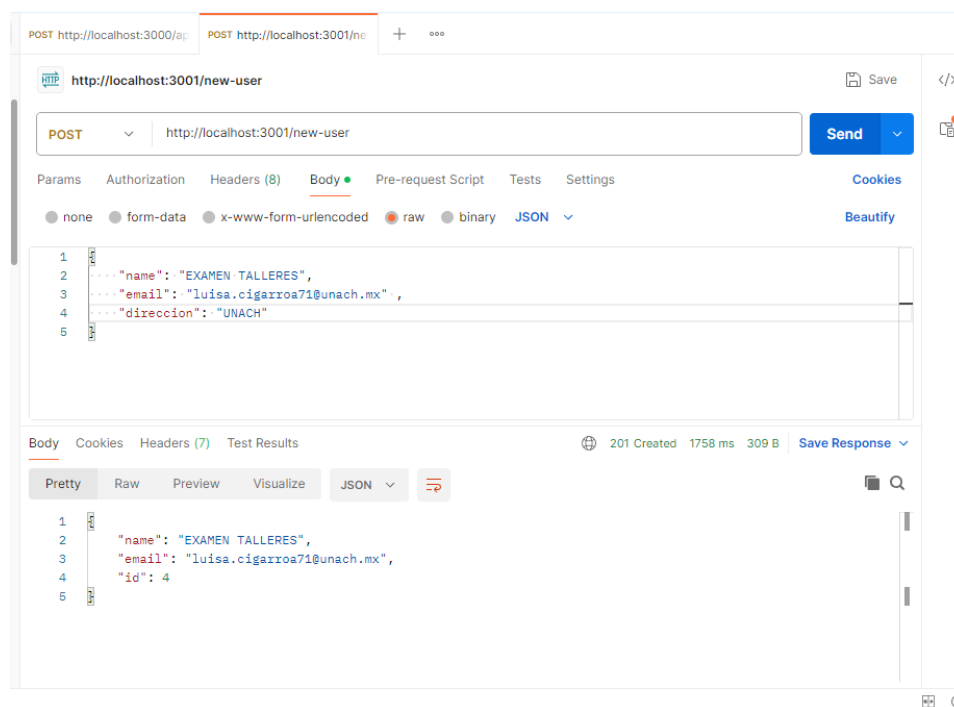
```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

[Info] Webpack is building your sources...

webpack 5.90.1 compiled successfully in 1986 ms
Type-checking in progress...
[Nest] 10660 - 16/04/2024, 7:29:00 p.m. LOG [NestFactory] Starting Nest application...
[Nest] 10660 - 16/04/2024, 7:29:00 p.m. LOG [InstanceLoader] AppModule dependencies initialized +17ms
[Nest] 10660 - 16/04/2024, 7:29:00 p.m. LOG [NestMicroservice] Nest microservice successfully started +72ms
No errors found.

1ms
[Nest] 14996 - 16/04/2024, 7:33:22 p.m. LOG [InstanceLoader] AppModule dependencies initialized +2ms
[Nest] 14996 - 16/04/2024, 7:33:22 p.m. LOG [RoutesResolver] AppController {/}: +10ms
[Nest] 14996 - 16/04/2024, 7:33:22 p.m. LOG [RouterExplorer] Mapped {/new-user, POST} route +5ms
[Nest] 14996 - 16/04/2024, 7:33:22 p.m. LOG [NestApplication] Nest application successfully started +3ms
No errors found.
```

EMPEZAMOS A INSERTAR DATOS



COMPROBAMOS QUE APAREZCA EN NUESTRA BASE DE DATOS

Result Grid			
Filter Rows:			
	id	name	email
1	1	Luisa	luisa.cigarroa71@unach.mx
2	2	Fernanda	luisa005hernandez@gmail.com
3	3	Profe Alfaro	alfaro2023.gmail.com
4	4	EXAMEN TALLERES	luisa.cigarroa71@unach.mx
	NULL	NULL	NULL

LINK DEL GIT

<https://github.com/luisafer200/EXAMEN-TALLERES-2-PARCIAL>