

# Universidad Autónoma de Chiapas | Campus 01 | Facultad de contaduría y administración.

Docente: Gutiérrez Alfaro Luis, Dr.

Materia: Taller de desarrollo 4.

Nombre del alumno (s):

- Carrasco Zavala Carlos Emmanuel | A210731.

Semestre: 6° | Grupo: "M".

Actividad-2.1. | AA .

Número de Actividad: Act.2.1 crear un DTO en Nest

Unidad 2

Tuxtla Gutiérrez, Chiapas; a 07 de Marzo del 2024.

GITHUB: <https://github.com/CarlosV01/DTO-Nest.git>

## Anexar Procedimiento: tomar las capturas de pantallas

GET

The screenshot displays the Visual Studio Code interface. On the left, the Explorer sidebar shows the project structure with files like `test.dto.ts`, `test.interface.ts`, and `test.controller.spec.ts`. The central editor shows the `test.controller.ts` file with the following code:

```
1 import { Controller, Delete, Get, Param, Patch, Post } from '@nestjs/common';
2 import { Request } from 'express';
3
4 export interface TestDto {
5   readonly description: string;
6   readonly isDone: boolean;
7 }
8
9 @Controller('projectX/test')
10 export class TestController {
11
12   @Get()
13   postMethod(@Req() req: Request): string {
14     return `Method ${req.method}`;
15   }
16
17   @Put()
18   putMethod(@Req() req: Request): string {
19     return `Taller ${req.method}`;
20   }
21
22   @Post('/:id/descripcion/:descripcion/cantidad/:cantidad')
23   Methodpost(@Param() Param: any) {
24     return {Param};
25   }
26 }
```

On the right, the REST client panel shows a GET request to `http://localhost:3000/projectX/test/`. The response is `200 OK` with a status of `7 ms` and a body of `238 B`. The terminal at the bottom shows logs from NestJS:

```
[Nest] 24556 - 07/03/2024, 8:46:07 a.m. LOG [InstanceLoader] TestController loaded
[Nest] 24556 - 07/03/2024, 8:46:07 a.m. LOG [InstanceLoader] TestController loaded
[Nest] 24556 - 07/03/2024, 8:46:07 a.m. LOG [RoutesResolver] TestController: /projectX/test
[Nest] 24556 - 07/03/2024, 8:46:07 a.m. LOG [RouterExplorer] Mapped {projectX/test, GET} to /projectX/test
[Nest] 24556 - 07/03/2024, 8:46:07 a.m. LOG [RouterExplorer] Mapped {projectX/test, PUT} to /projectX/test
[Nest] 24556 - 07/03/2024, 8:46:07 a.m. LOG [RouterExplorer] Mapped {projectX/test/:id/descripcion/:descripcion/cantidad/:cantidad, POST} to /projectX/test/:id/descripcion/:descripcion/cantidad/:cantidad
```

GITHUB: <https://github.com/CarlosV01/DTO-Nest.git>

POST

The image shows a development environment with VS Code on the left and Postman on the right.

**VS Code:**

- EXPLORADOR:** Shows the project structure with folders like `dist`, `node_modules`, `src`, and `test`. The `test` folder contains `dto`, `test.dto.ts`, `test.interface.ts`, `test.controller.spec.ts`, `test.controller.ts`, `test.module.ts`, `test.service.spec.ts`, `test.service.ts`, `app.controller.spec.ts`, `app.controller.ts`, `app.module.ts`, `app.service.ts`, `index.html`, and `main.ts`.
- EDITOR:** Displays the `test.controller.ts` file. The code defines a `TestDto` interface and a `TestController` class with methods `getMethod`, `putMethod`, and `postMethod`.
- TERMINAL:** Shows the output of the `npm run test` command, indicating that the tests passed.

**Postman:**

- Overview:** Shows the API endpoint `http://localhost:3000/projectX/test/yo/description/tall/cantid...` with a `POST` method.
- Params:** Shows the query parameters table.
- Path Variables:** Shows the path variables table.
- Body:** Shows the request body in JSON format.

GITHUB: <https://github.com/CarlosV01/DTO-Nest.git>

PUT

The image shows a development environment with VS Code on the left and Postman on the right.

**VS Code (Left):**

- EXPLORADOR:** Shows the project structure with folders like `dist`, `node_modules`, `src`, and `test`. The `test` folder contains `dto`, `test.dto.ts`, `test.interface.ts`, `test.controller.spec.ts`, `test.service.ts`, `app.controller.spec.ts`, `app.controller.ts`, `app.module.ts`, `app.service.ts`, `index.html`, and `main.ts`.
- EDITOR:** Displays the `test.controller.ts` file. The code defines a `TestController` class with methods `@Get()`, `@Put()`, `@Post()`, `@Patch()`, and `@Delete()`. The `@Put()` method is highlighted.
- TERMINAL:** Shows the output of the application, including logs for `RouterExp` and `NestApp`.

**Postman (Right):**

- Overview:** Shows the request URL `http://localhost:3000/projectX/test/` and the method `PUT`.
- Params:** Shows the query parameters table.
- Body:** Shows the response body in `Pretty` format, displaying `1 DONE PUT`.