



“UNIVERSIDAD DE LAS FUERZAS ARMADAS”

ESPE

DEPARTAMENTO CIENCIAS DE COMPUTACIÓN

CARRERA DE INGENIERÍA DE SOFTWARE

ASIGNATURA: DESARROLLO WEB AVANZADO

TEMA: EVALUACION

GRUPO 2

INTEGRANTES

RICHARD STEVE ALBAN VARGAS

RICARDO ANDRÉ LAZO TORRES

CARLOS JOAO IPIALES ULCUANGO

TUTOR: ING. DIEGO MEDARDO SAAVEDRA GARCIA

FECHA: 07/03/2024

Link del github: <https://github.com/Rich4rd16/evaluation-ghostshell-grupo4.git>

Funcionamiento

Debemos considerar los siguientes pasos para el correcto funcionamiento de la evaluación.

1) Paso

En la raíz de la evaluación, ejecutaremos el siguiente comando para instalar todas las dependencias.

“npm install”

En caso de haber algún error estas son las dependencias que contiene la evaluación.

```
"dependencies": {  
  "@nestjs/common": "^10.0.0",  
  "@nestjs/core": "^10.0.0",  
  "@nestjs/mongoose": "^10.0.4",  
  "@nestjs/platform-express": "^10.0.0",  
  "bson": "^6.4.0",  
  "mongoose": "^8.2.0",  
  "reflect-metadata": "^0.2.0",  
  "rxjs": "^7.8.1"  
},
```

Ejecutamos el siguiente comando para instalar todas las dependencias de la evaluación.


“npm install @nestjs/common@^10.0.0 @nestjs/core@^10.0.0 @nestjs/mongoose@^10.0.4 @nestjs/platform-express@^10.0.0 bson@^6.4.0 mongoose@^8.2.0 reflect-metadata@^0.2.0 [rxjs@^7.8.1](#)”

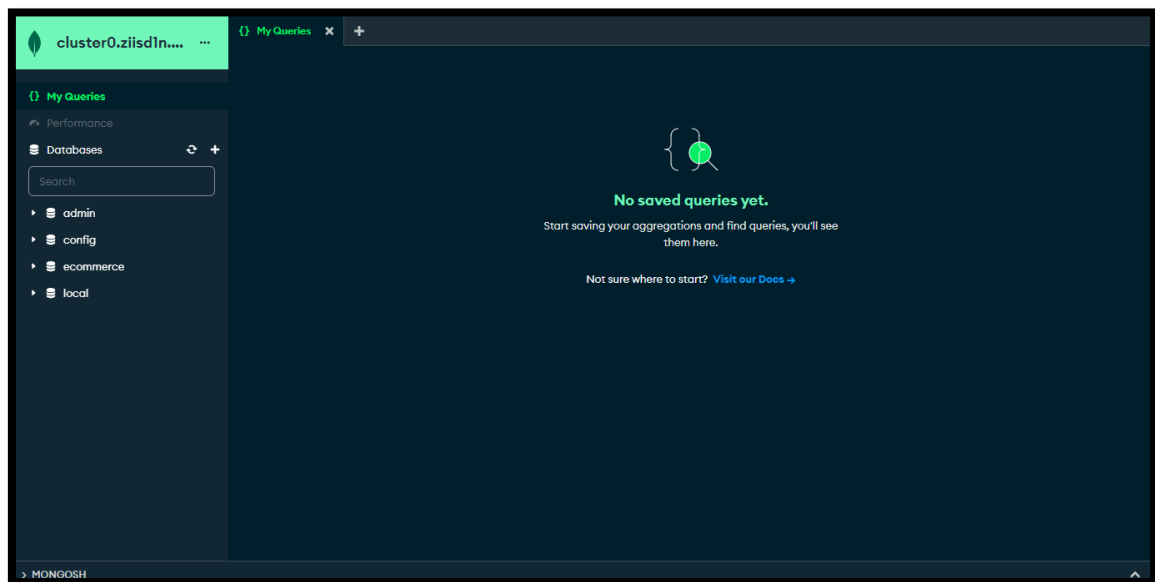
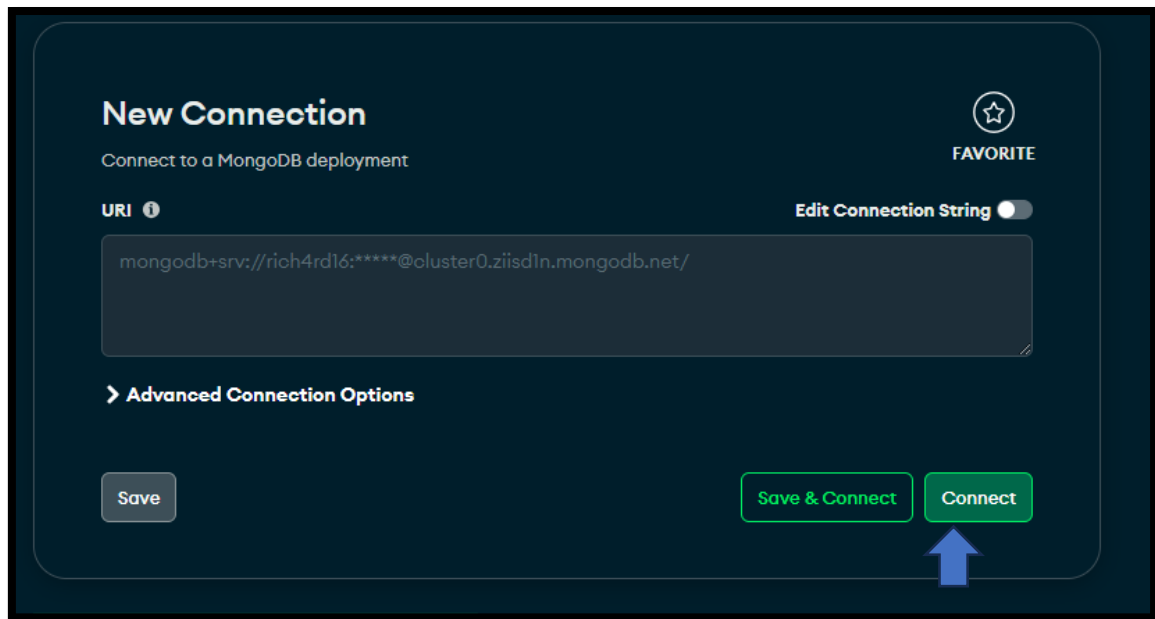
2) Paso

Abrimos nuestra base de datos MongoCompass, con la url asignada en el archivo *“app.module.ts”*.

“mongodb+srv://rich4rd16:ESPE178059@cluster0.ziisd1n.mongodb.net/ecommerce”

```
@Module({  
  imports: [  
    MongooseModule.forRoot('mongodb+srv://rich4rd16:ESPE178059@cluster0.ziisd1n.mongodb.net/ecommerce'),  
    CursosModule,  
    TemasModule,  
  ],  
})  
export class AppModule {}
```





3) Paso

Ejecutamos la siguiente línea de comando para iniciar nuestro proyecto de evaluación.

“npm start”

```
> nest start

[Nest] 16640 - 07/03/2024, 22:07:49 LOG [NestFactory] Starting Nest application...
[Nest] 16640 - 07/03/2024, 22:07:49 LOG [InstanceLoader] AppModule dependencies initialized +39ms
[Nest] 16640 - 07/03/2024, 22:07:49 LOG [InstanceLoader] MongooseModule dependencies initialized +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [InstanceLoader] MongooseCoreModule dependencies initialized +1682ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [InstanceLoader] MongooseModule dependencies initialized +30ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [InstanceLoader] MongooseModule dependencies initialized +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [InstanceLoader] CursosModule dependencies initialized +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [InstanceLoader] TemasModule dependencies initialized +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RoutesResolver] CursosController {/cursos}: +38ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/cursos, POST} route +7ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/cursos, GET} route +2ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/cursos/:id, GET} route +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/cursos/:id, PUT} route +2ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/cursos/:id, DELETE} route +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/cursos/:cursoId/temas/:temaId, POST} route +2ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/cursos/:cursoId/temas/:temaId, DELETE} route +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RoutesResolver] TemasController {/temas}: +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/temas, POST} route +2ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/temas, GET} route +3ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/temas/:id, GET} route +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/temas/:id, PUT} route +1ms
[Nest] 16640 - 07/03/2024, 22:07:51 LOG [RouterExplorer] Mapped {/temas/:id, DELETE} route +1ms
```

4) Paso

Procedemos a realizar el CRUD en cursos y temas, además de la relación de un curso puede tener varios temas.

- **Cursos**

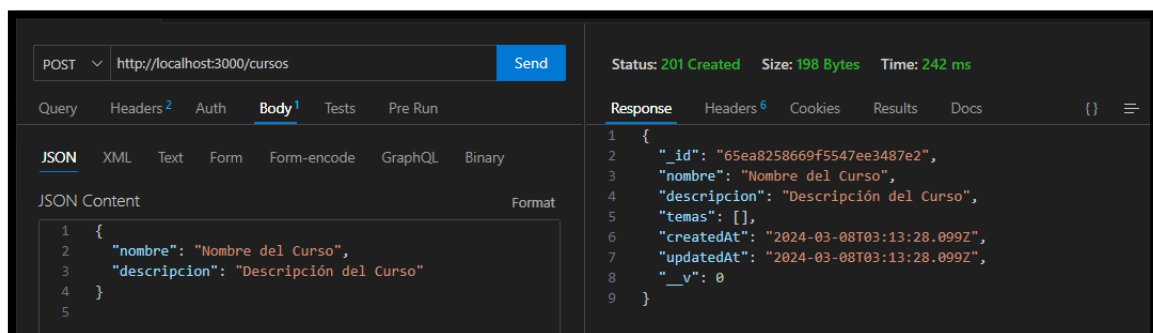
- **Método Post**

Ruta: <http://localhost:3000/cursos>

JSON content:

```
{
  "nombre": "Nombre del Curso",
  "descripcion": "Descripción del Curso"
}
```

Visualización en ThunderClient



Visualización en la base de datos MongoDB (Se va a la colección cursos).



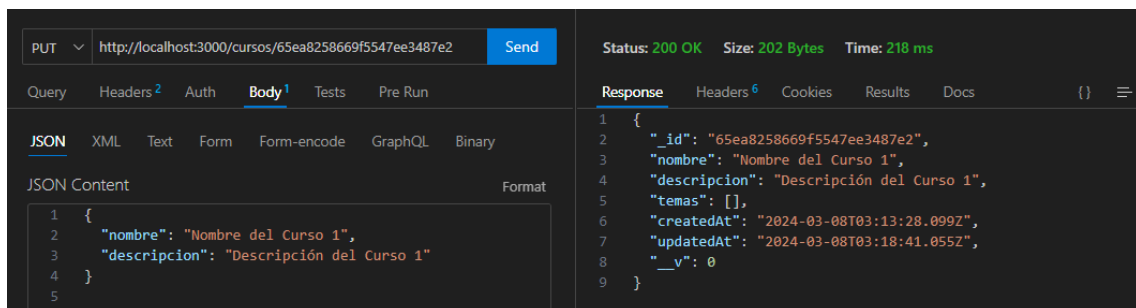
➤ Método Put

Ruta: <http://localhost:3000/cursos/65ea8258669f5547ee3487e2>

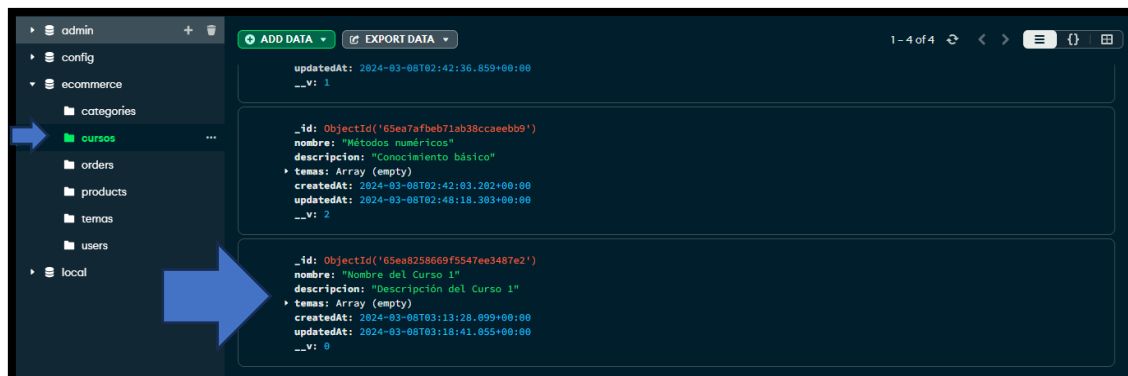
JSON content:

```
{
  "nombre": "Nombre del Curso 1",
  "descripcion": "Descripción del Curso 1"
}
```

Visualización en ThunderClient



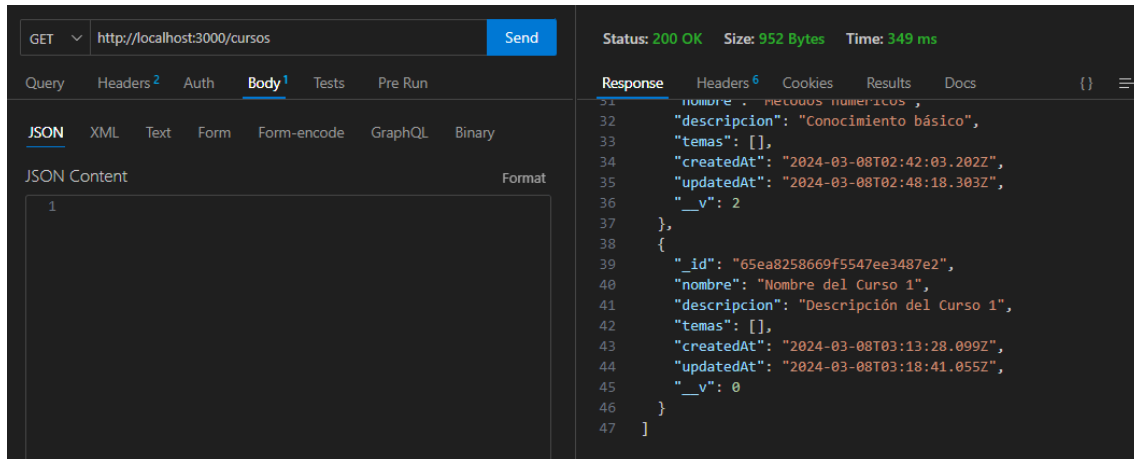
Visualización en la base de datos MongoDB (Se va a la colección cursos).



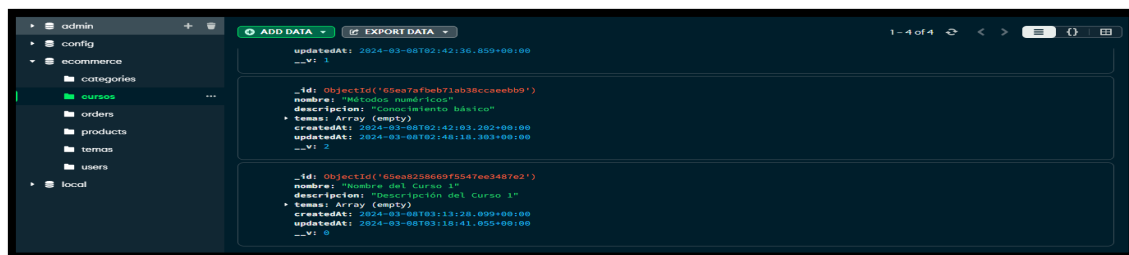
➤ Método Get

Ruta: <http://localhost:3000/cursos>

Visualización en ThunderClient

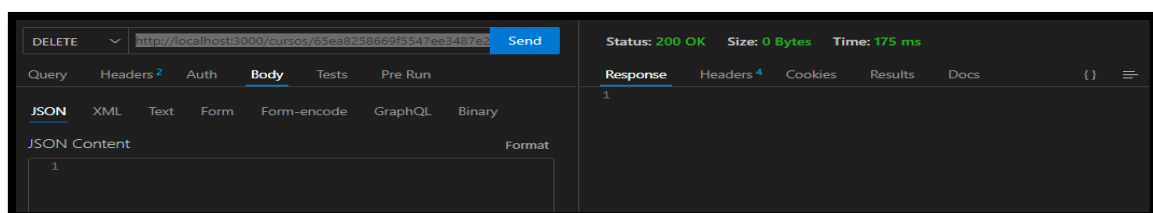


Visualización en la base de datos MongoDB (Se va a la colección cursos).

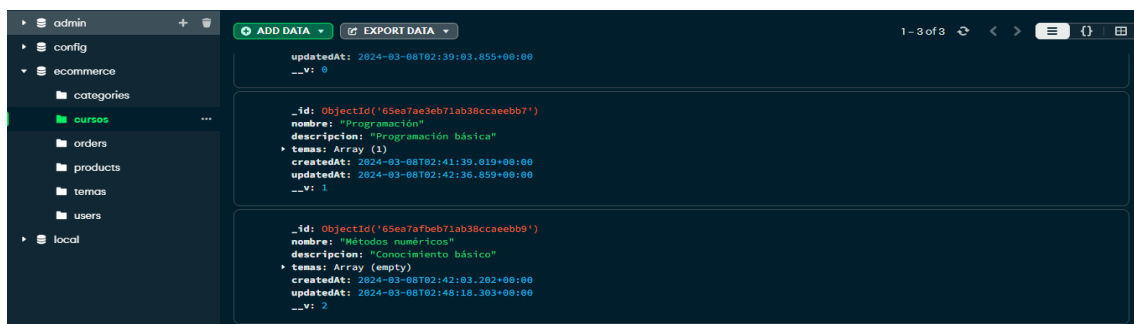


➤ Método Delete

Ruta: <http://localhost:3000/cursos/65ea8258669f5547ee3487e2>



Visualización en la base de datos MongoDB (Se observa que se eliminó el dato).



- **Temas**

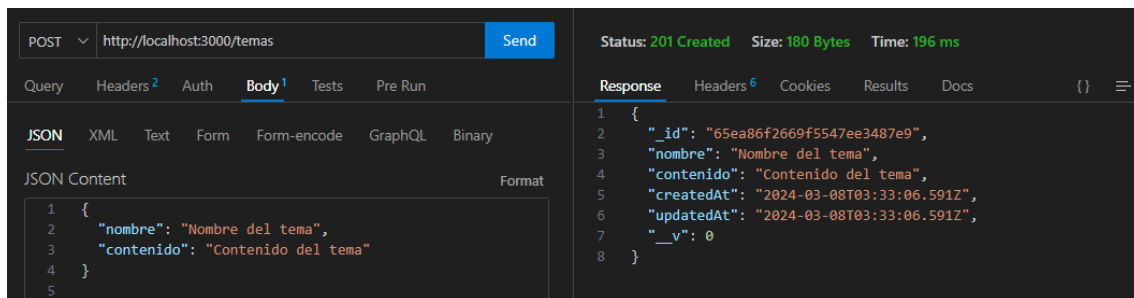
- **Método Post**

Ruta: <http://localhost:3000/temas>

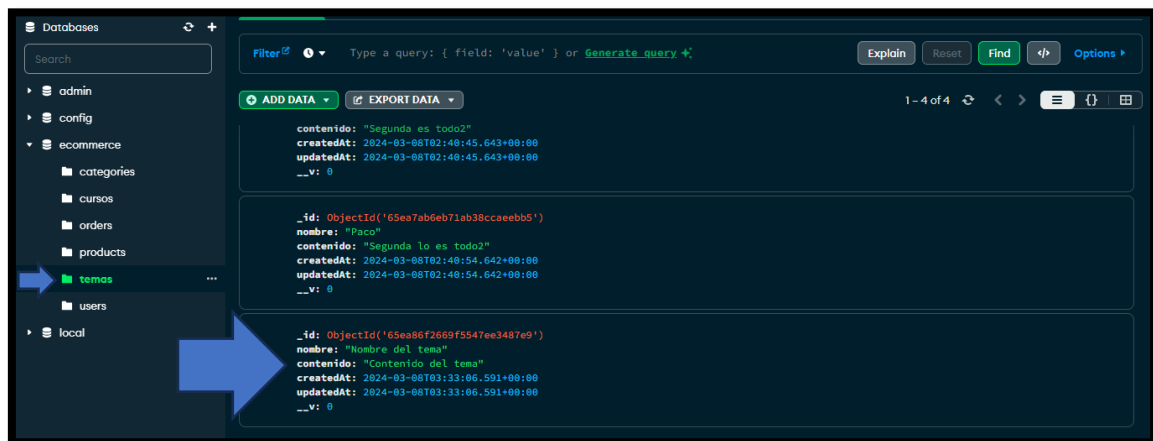
JSON content:

```
{
  "nombre": "Nombre del tema",
  "contenido": "Contenido del tema"
}
```

Visualización en ThunderClient



Visualización en la base de datos MongoDB (Se va a la colección temas).



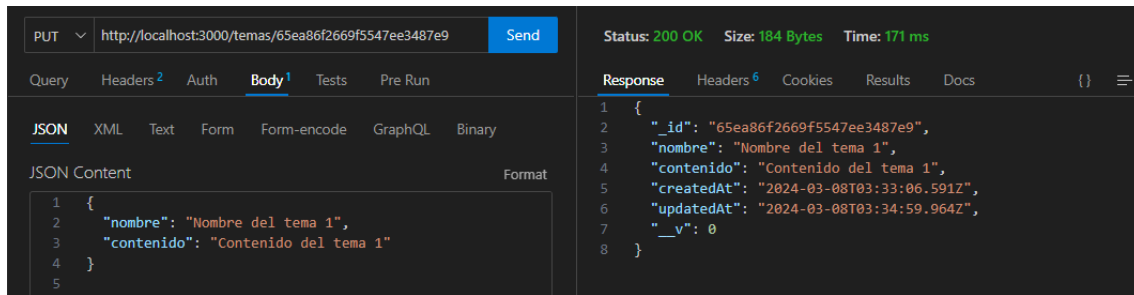
- **Método Put**

Ruta: <http://localhost:3000/cursos/65ea8258669f5547ee3487e2>

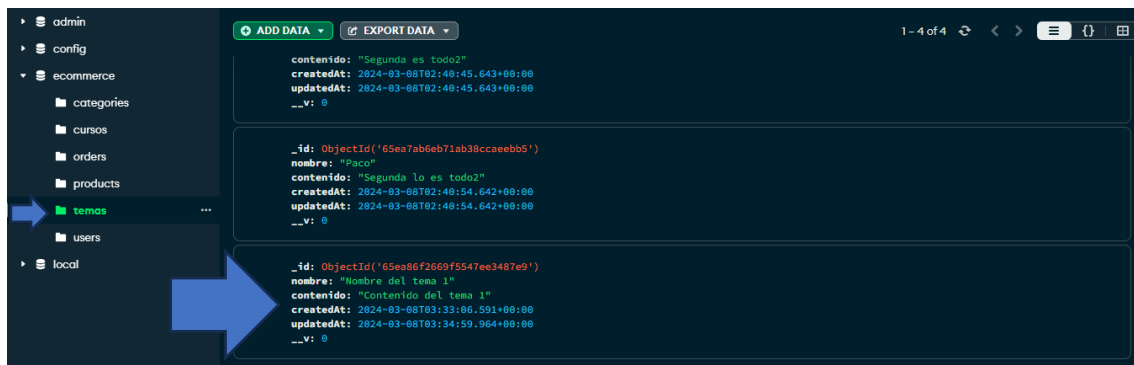
JSON content:

```
{
  "nombre": "Nombre del Curso 1",
  "descripcion": "Descripción del Curso 1"
}
```

Visualización en ThunderClient



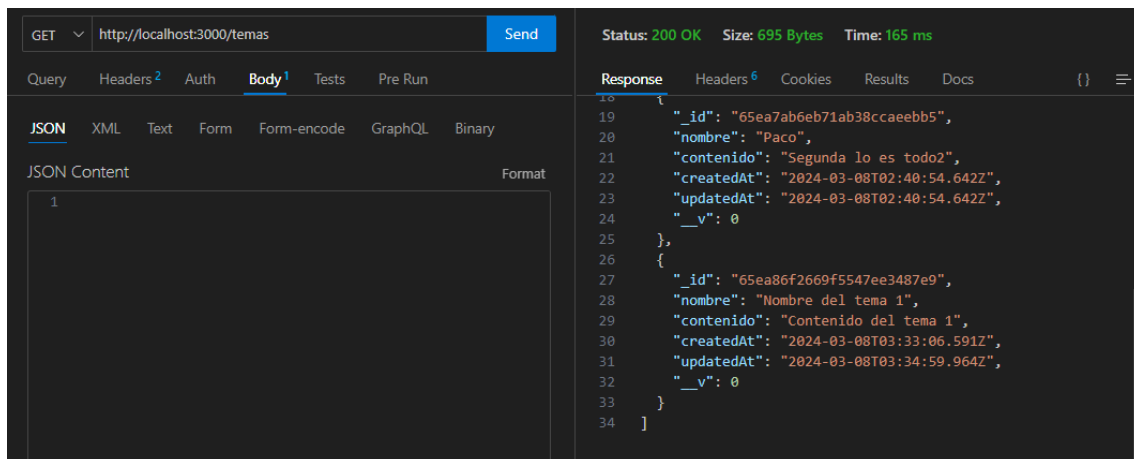
Visualización en la base de datos MongoDB (Se va a la colección temas).



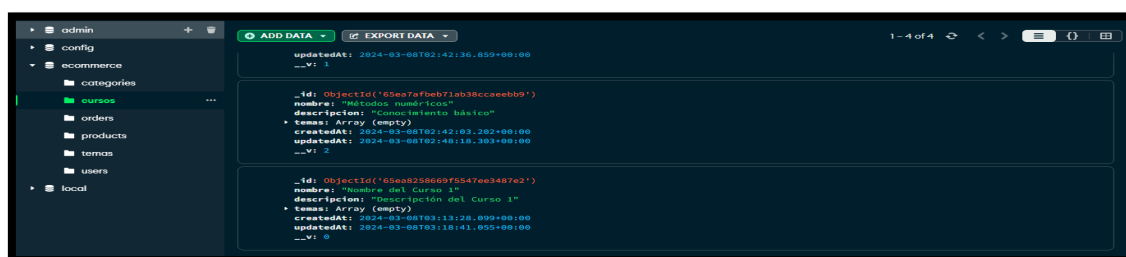
➤ Método Get

Ruta: <http://localhost:3000/temas/65ea86f2669f5547ee3487e9>

Visualización en ThunderClient



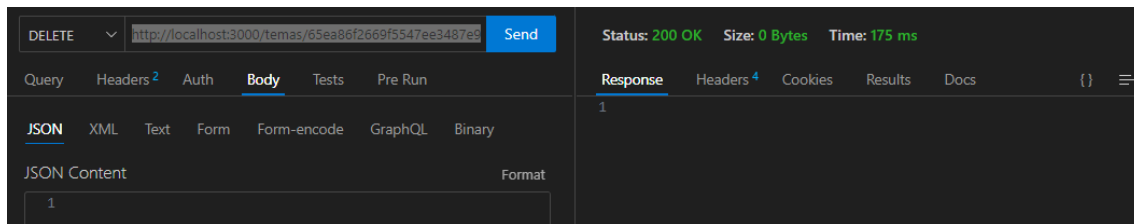
Visualización en la base de datos MongoDB (Se va a la colección cursos).



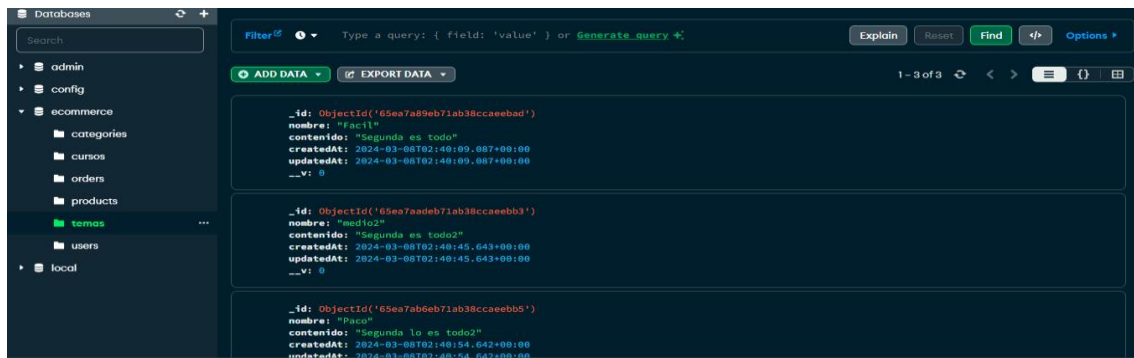
➤ Método Delete

Ruta: <http://localhost:3000/temas/65ea86f2669f5547ee3487e9>

Visualización en ThunderClient



Visualización en la base de datos MongoDB (Se observa que se eliminó el dato).

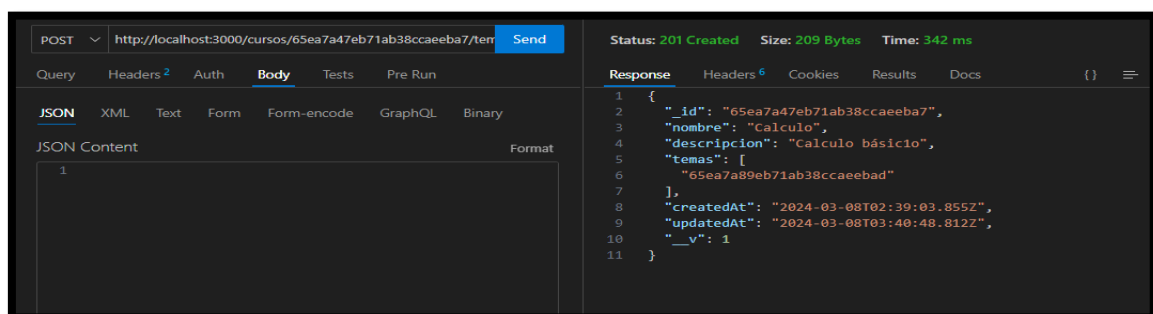


- Asignación y eliminación de un tema a un curso

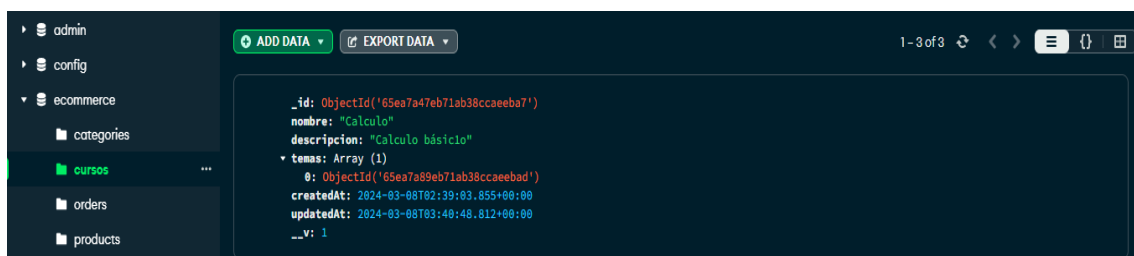
➤ Utilizamos la opción Post

Ruta: <http://localhost:3000/cursos/65ea7a47eb71ab38ccaeeba7/temas/65ea7a89eb71ab38ccaeebad>

Visualización en ThunderClient

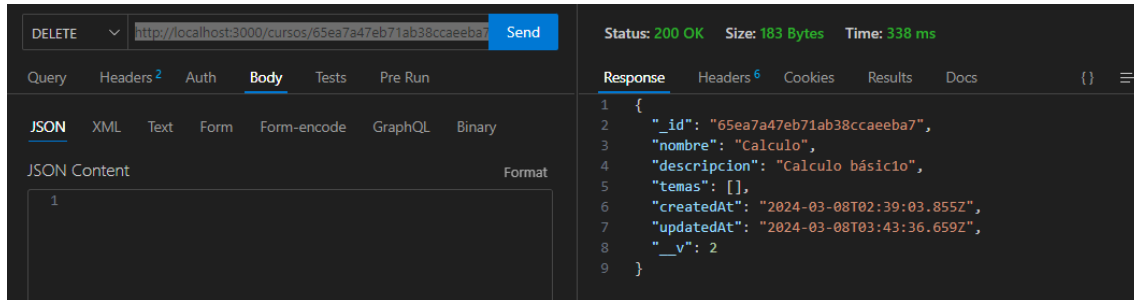


Visualización en la base de datos MongoDB (En la colección cursos).



- Utilizamos la opción Delete
Ruta: <http://localhost:3000/cursos/65ea7a47eb71ab38ccaeba7/temas/65ea7a89eb71ab38ccaebad>

Visualización en ThunderClient



Visualización en la base de datos MongoDB (En la colección cursos).

