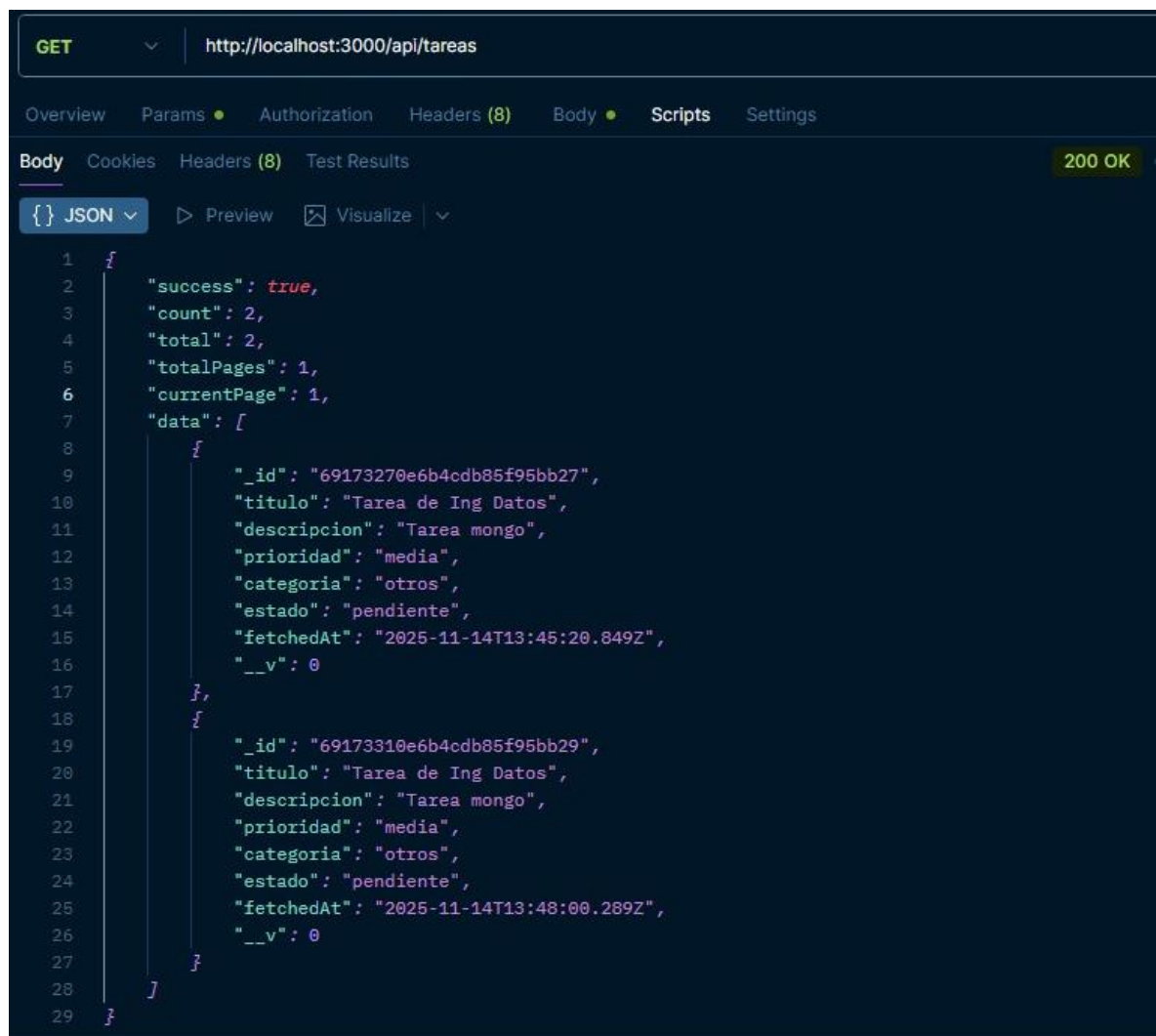


Para esta ultima actividad relacionada con el proceso express para relacionar bases de datos con API´s se procedió a hacer las debidas pruebas sobre las tareas que fueron definidas en el script. Las cuales son:

```
1  {
2    "message": "🔥 API de Gestión de Tareas",
3    "version": "1.0.0",
4    "endpoints": {
5      "tareas": {
6        "obtenerTodas": "GET /api/tareas",
7        "obtenerPorId": "GET /api/tareas/:id",
8        "crear": "POST /api/tareas",
9        "actualizar": "PUT /api/tareas/:id",
10       "eliminar": "DELETE /api/tareas/:id",
11       "completar": "PATCH /api/tareas/:id/completar",
12       "reabrir": "PATCH /api/tareas/:id/reabrir",
13       "estadisticas": "GET /api/tareas/stats",
14       "porPrioridad": "GET /api/tareas/prioridad/:prioridad"
15     }
16   },
17 }
```

1. obtenerTodas: Tareas que estén guardadas dentro de la base de datos.

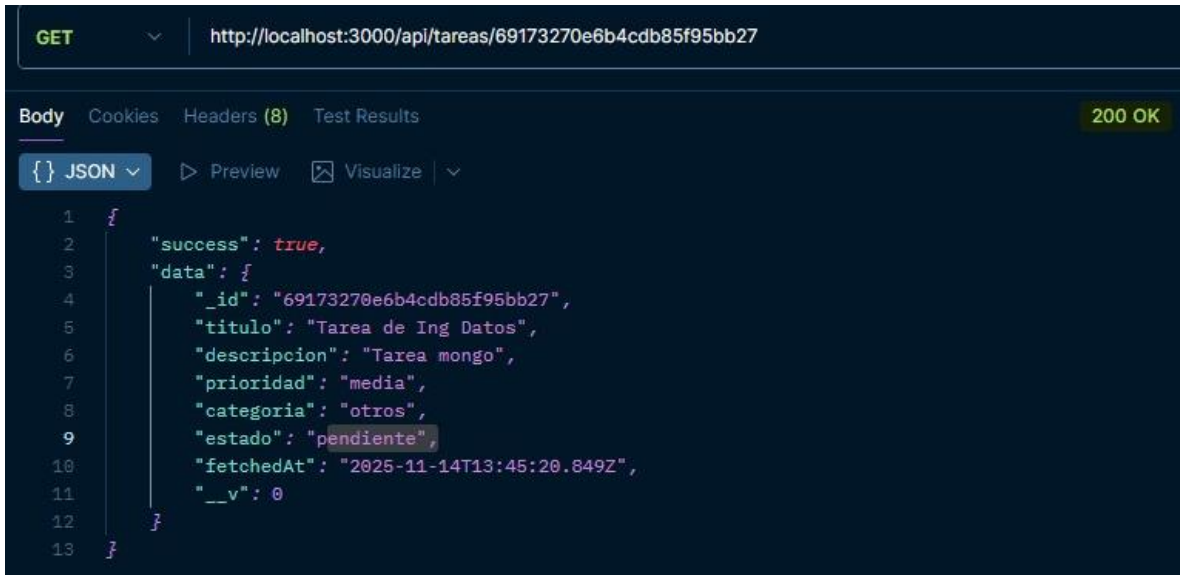


The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:3000/api/tareas
- Status:** 200 OK
- Body:** JSON

```
1  {
2    "success": true,
3    "count": 2,
4    "total": 2,
5    "totalPages": 1,
6    "currentPage": 1,
7    "data": [
8      {
9        "_id": "69173270e6b4cdb85f95bb27",
10       "titulo": "Tarea de Ing Datos",
11       "descripcion": "Tarea mongo",
12       "prioridad": "media",
13       "categoria": "otros",
14       "estado": "pendiente",
15       "fetchedAt": "2025-11-14T13:45:20.849Z",
16       "__v": 0
17     },
18     {
19       "_id": "69173310e6b4cdb85f95bb29",
20       "titulo": "Tarea de Ing Datos",
21       "descripcion": "Tarea mongo",
22       "prioridad": "media",
23       "categoria": "otros",
24       "estado": "pendiente",
25       "fetchedAt": "2025-11-14T13:48:00.289Z",
26       "__v": 0
27     }
28   ]
29 }
```

## 2. obtenerPorId: Consultar tareas guardadas por número de id.



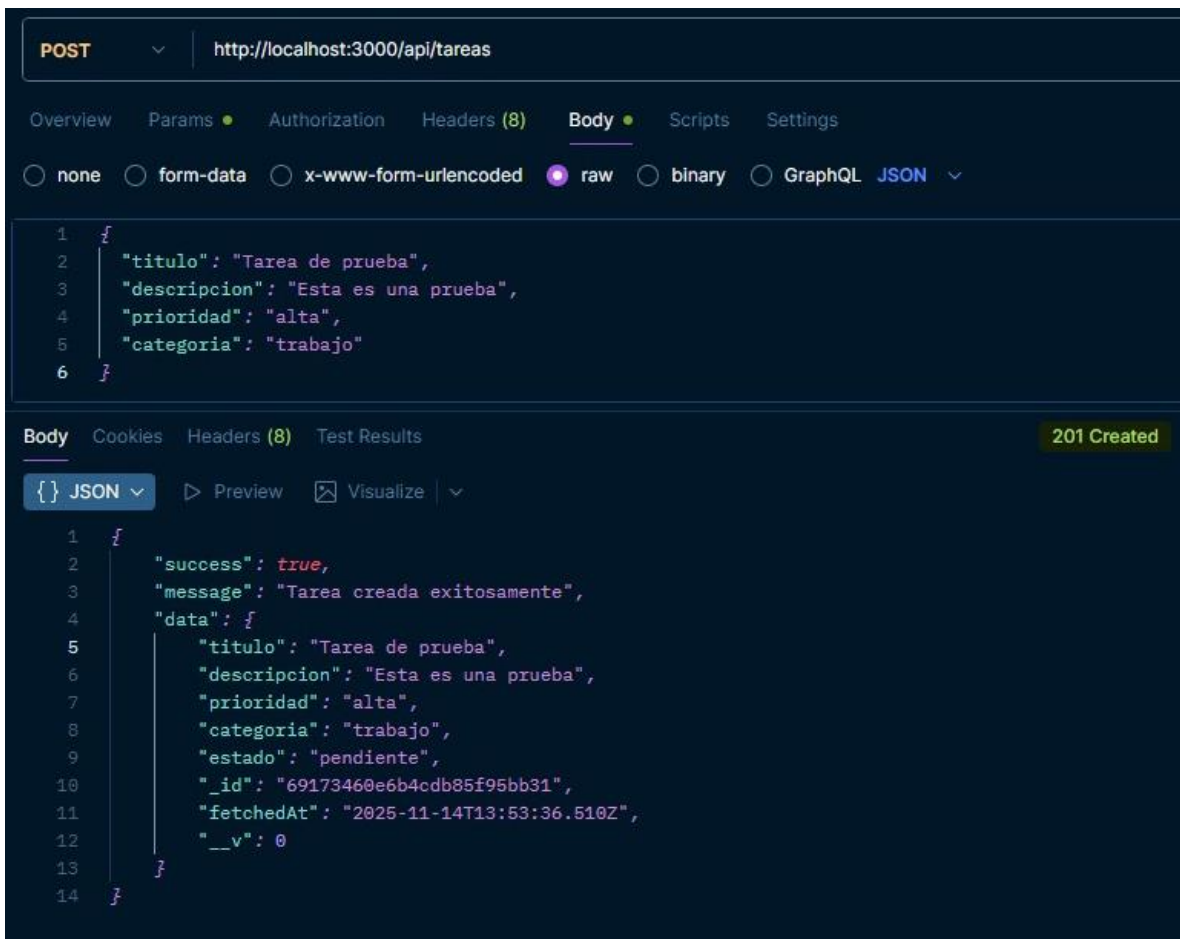
```
GET http://localhost:3000/api/tareas/69173270e6b4cdb85f95bb27

Body Cookies Headers (8) Test Results 200 OK

JSON Preview Visualize

1 {
2   "success": true,
3   "data": {
4     "_id": "69173270e6b4cdb85f95bb27",
5     "titulo": "Tarea de Ing Datos",
6     "descripcion": "Tarea mongo",
7     "prioridad": "media",
8     "categoria": "otros",
9     "estado": "pendiente",
10    "fetchAt": "2025-11-14T13:45:20.849Z",
11    "__v": 0
12  }
13 }
```

## 3. crear: Encargada de indexar nueva información a la base de datos.



```
POST http://localhost:3000/api/tareas

Overview Params Authorization Headers (8) Body Scripts Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {
2   "titulo": "Tarea de prueba",
3   "descripcion": "Esta es una prueba",
4   "prioridad": "alta",
5   "categoria": "trabajo"
6 }

Body Cookies Headers (8) Test Results 201 Created

JSON Preview Visualize

1 {
2   "success": true,
3   "message": "Tarea creada exitosamente",
4   "data": {
5     "titulo": "Tarea de prueba",
6     "descripcion": "Esta es una prueba",
7     "prioridad": "alta",
8     "categoria": "trabajo",
9     "estado": "pendiente",
10    "_id": "69173460e6b4cdb85f95bb31",
11    "fetchAt": "2025-11-14T13:53:36.510Z",
12    "__v": 0
13  }
14 }
```

4. actualizar: Modificar una tarea ya registrada y guardarla.

The screenshot shows a REST client interface with a PUT request to `http://localhost:3000/api/tareas/69173270e6b4cdb85f95bb27`. The request body is a JSON object with the following fields:

```
1 {
2   "titulo": "Comprar pan y leche y un tamal",
3   "descripcion": "Panadería y supermercado",
4   "prioridad": "alta"
5 }
6
```

The response body is also shown in JSON format:

```
1 {
2   "success": true,
3   "message": "Tarea actualizada exitosamente",
4   "data": {
5     "_id": "69173270e6b4cdb85f95bb27",
6     "titulo": "Comprar pan y leche y un tamal",
7     "descripcion": "Panadería y supermercado",
8     "prioridad": "alta",
9     "categoria": "otros",
10    "estado": "pendiente",
11    "fetchAt": "2025-11-14T13:45:20.849Z",
12    "__v": 0
13  }
14 }
```

5. eliminar: Borra un registro de la base de datos.

The screenshot shows a REST client interface with a DELETE request to `http://localhost:3000/api/tareas/69173270e6b4cdb85f95bb27`. The request body is a JSON object with the following fields:

```
1 {
2   "titulo": "Comprar pan y leche y un tamal",
3   "descripcion": "Panadería y supermercado",
4   "prioridad": "alta"
5 }
6
```

The response body is also shown in JSON format:

```
1 {
2   "success": true,
3   "message": "Tarea eliminada exitosamente",
4   "data": {}
5 }
```

6. completar: Cambia el estado de una tarea o registro a “completada”.

The screenshot shows a REST client interface with a PATCH request to `http://localhost:3000/api/tareas/6917348fe6b4cdb85f95bb33/completar`. The request body is a JSON object with the following fields:

```
1 {
2   "titulo": "Comprar pan y leche y un tamal",
3   "descripcion": "Panadería y supermercado",
4   "prioridad": "alta"
5 }
6
```

The response body is also shown in JSON format:

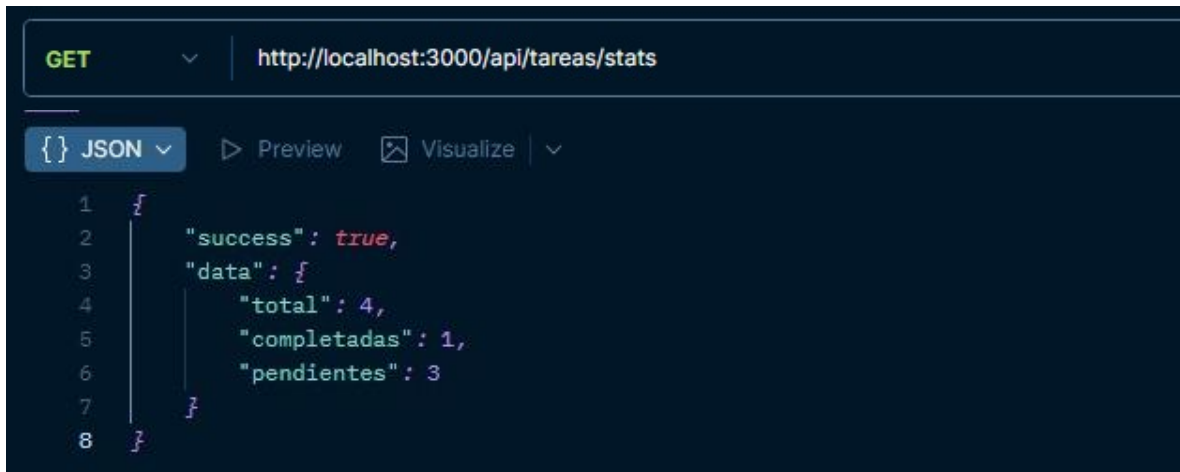
```
1 {
2   "success": true,
3   "message": "Tarea completada",
4   "data": {
5     "_id": "6917348fe6b4cdb85f95bb33",
6     "titulo": "Tarea de prueba",
7     "descripcion": "Esta es una prueba",
8     "prioridad": "alta",
9     "categoria": "trabajo",
10    "estado": "completada",
11    "fetchAt": "2025-11-14T13:54:23.850Z",
12    "__v": 0
13  }
14 }
```

7. reabrir: Vuelve a cambiar el estado de una tarea o registro a “pendiente”.

The screenshot shows a REST client interface with a PATCH request to `http://localhost:3000/api/tareas/6917348fe6b4cdb85f95bb33/reabrir`. The request body is a JSON object with the following fields:

```
1 {
2   "success": true,
3   "message": "Tarea reabierta",
4   "data": {
5     "_id": "6917348fe6b4cdb85f95bb33",
6     "titulo": "Tarea de prueba",
7     "descripcion": "Esta es una prueba",
8     "prioridad": "alta",
9     "categoria": "trabajo",
10    "estado": "pendiente",
11    "fetchAt": "2025-11-14T13:54:23.850Z",
12    "__v": 0
13  }
14 }
```

8. estadísticas: Devuelve información de las tareas de las cuales están: completadas, total y pendientes.

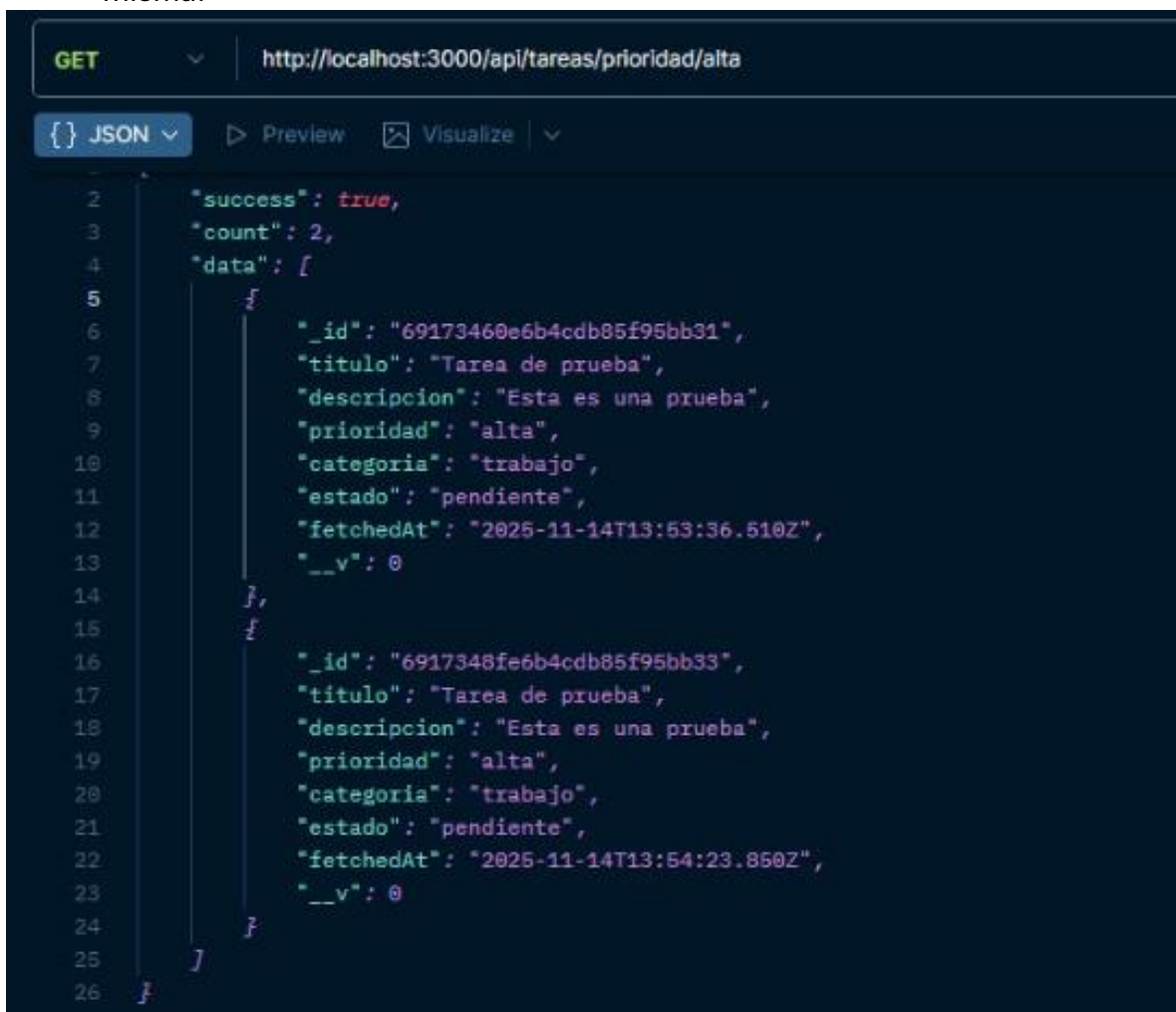


The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: `http://localhost:3000/api/tareas/stats`
- Response Format: JSON
- Response Body (lines 1-8):

```
1 {
2   "success": true,
3   "data": {
4     "total": 4,
5     "completadas": 1,
6     "pendientes": 3
7   }
8 }
```

9. porPrioridad: Me filtra por prioridad “alta” las tareas que tengan asignada la misma.



The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: `http://localhost:3000/api/tareas/prioridad/alta`
- Response Format: JSON
- Response Body (lines 2-26):

```
2   "success": true,
3   "count": 2,
4   "data": [
5     {
6       "_id": "69173460e6b4cdb85f95bb31",
7       "titulo": "Tarea de prueba",
8       "descripcion": "Esta es una prueba",
9       "prioridad": "alta",
10      "categoria": "trabajo",
11      "estado": "pendiente",
12      "fetchAt": "2025-11-14T13:53:36.510Z",
13      "__v": 0
14    },
15    {
16      "_id": "6917348fe6b4cdb85f95bb33",
17      "titulo": "Tarea de prueba",
18      "descripcion": "Esta es una prueba",
19      "prioridad": "alta",
20      "categoria": "trabajo",
21      "estado": "pendiente",
22      "fetchAt": "2025-11-14T13:54:23.850Z",
23      "__v": 0
24    }
25  ]
26 }
```