

resultado de los 10 TEST de registro

POST Producto:

1)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/producto/registerProducto`. The request body is a JSON object representing a product. The response body shows the server's reply, which includes a unique ID, the product details, a timestamp, and a version number.

Request:

```
1  {
2    "name": "Hamburguesa",
3    "precio": "10000",
4    "codigo": "111",
5    "description": "Doble carne"
6  }
```

Response:

```
1  {
2    "producto": {
3      "_id": "611303625a904b0e6c6c8a1f",
4      "name": "Hamburguesa",
5      "precio": "10000",
6      "codigo": "111",
7      "description": "Doble carne",
8      "fecha": "2021-08-10T22:53:22.029Z",
9      "__v": 0
10   }
11 }
```

2)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/producto/registerProducto`. The request body is a JSON object with the following fields:

```
{  "name": "Ajiaco",  "precio": "2000",  "codigo": "451",  "description": "Re caliente"}
```

The response body is also in JSON format, showing the registered product details:

```
{  "producto": {    "_id": "611303af5a904b0e6c6c8a22",    "name": "Ajiaco",    "precio": "2000",    "codigo": "451",    "description": "Re caliente",    "fecha": "2021-08-10T22:54:39.801Z",    "__v": 0  }}
```

3)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/producto/registerProducto`. The request body is a JSON object with the following fields:

```
{
  "name": "Arepa rellena",
  "precio": "6000",
  "codigo": "147",
  "description": "Super baratas"
}
```

The response is shown in the 'Body' tab, formatted as JSON:

```
{
  "producto": {
    "_id": "611304145a904b0e6c6c8a25",
    "name": "Arepa rellena",
    "precio": "6000",
    "codigo": "147",
    "description": "Super baratas",
    "fecha": "2021-08-10T22:56:20.632Z",
    "__v": 0
  }
}
```

4)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/producto/registerProducto`. The request body is a JSON object with the following fields:

```
1 {
2   "name": "Salchipapas",
3   "precio": "8000",
4   "codigo": "159",
5   "description": "muchos tamaños"
6 }
```

Below the request, the response body is displayed in a 'Pretty' JSON format:

```
1 {
2   "producto": {
3     "_id": "611304375a904b0e6c6c8a28",
4     "name": "Salchipapas",
5     "precio": "8000",
6     "codigo": "159",
7     "description": "muchos tamaños",
8     "fecha": "2021-08-10T22:56:55.219Z",
9     "__v": 0
10  }
11 }
```

5)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/producto/registerProducto`. The request body is a JSON object with the following fields: `name` (Hotdog), `precio` (3000), `codigo` (153), and `description` (Con muchas salsas). The response body is a JSON object containing the product details, including a unique `_id`, the same product information, a timestamp `fecha`, and a version number `__v`.

POST `http://localhost:3001/api/producto/registerProducto`

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▾

```
1 {
2   "name": "Hotdog",
3   "precio": "3000",
4   "codigo": "153",
5   "description": "Con muchas salsas"
6 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "producto": {
3     "_id": "611304615a904b0e6c6c8a2b",
4     "name": "Hotdog",
5     "precio": "3000",
6     "codigo": "153",
7     "description": "Con muchas salsas",
8     "fecha": "2021-08-10T22:57:37.320Z",
9     "__v": 0
10  }
11 }
```

POST Almacén:

6)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/almacen/registerAlmacen`. The request body is a JSON object with the following fields:

```
{
  "name": "SuperEsquina",
  "direccion": "Cra 68 No 56-45",
  "ciudad": "Bogota"
}
```

The response is also in JSON format, showing the created record with an ID and a version number:

```
{
  "almacen": {
    "_id": "611320d81a96ee2488f3ec39",
    "name": "SuperEsquina",
    "direccion": "Cra 68 No 56-45",
    "ciudad": "Bogota",
    "__v": 0
  }
}
```

The interface includes tabs for Params, Authorization, Headers (8), Body, Pre-request Script, Tests, and Settings. The Body tab is active, showing the request and response in JSON format. The response is displayed in a 'Pretty' view.

7)

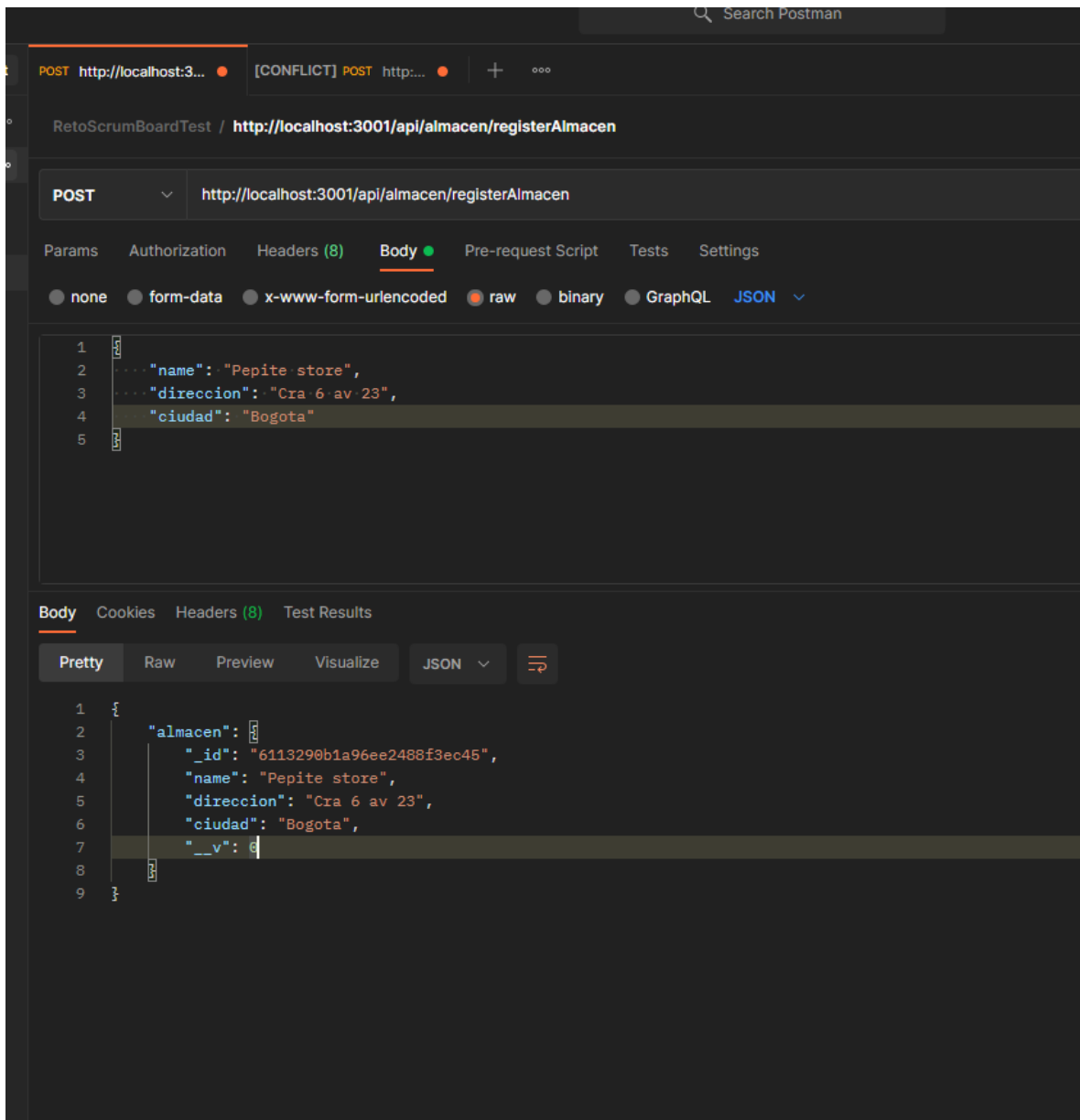
The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/almacen/registerAlmacen`. The request body is a JSON object with the following fields:

```
1 {
2   "name": "La tienda Moderna",
3   "direccion": "Cra 8 av 7",
4   "ciudad": "Risaralda"
5 }
```

The response is also in JSON format, showing the stored data with an assigned ID and a version number:

```
1 {
2   "almacen": {
3     "_id": "611323e61a96ee2488f3ec42",
4     "name": "La tienda Moderna",
5     "direccion": "Cra 8 av 7",
6     "ciudad": "Risaralda",
7     "__v": 0
8   }
9 }
```

8)



9)

The screenshot shows a REST client interface with a dark theme. At the top, there's a tab bar with a selected tab labeled "POST http://localhost:3001/api/almacen/registerAlmacen" and another tab labeled "[CONFLICT] POST http:...". Below the tab bar, the URL "http://localhost:3001/api/almacen/registerAlmacen" is displayed. The main interface has several tabs: "Params", "Authorization", "Headers (8)", "Body" (which is selected), "Pre-request Script", "Tests", and "Settings". Under the "Body" tab, there are radio buttons for "none", "form-data", "x-www-form-urlencoded", "raw" (which is selected), "binary", "GraphQL", and "JSON". The body content is a JSON object:

```
{ 1: { 2:   "name": "store pepis", 3:   "direccion": "Av 1", 4:   "ciudad": "Cali" 5: }
```

. Below the body tab, there are tabs for "Body", "Cookies", "Headers (8)", and "Test Results". The "Body" tab is selected, and it has sub-tabs: "Pretty" (selected), "Raw", "Preview", "Visualize", "JSON" (with a dropdown arrow), and a "Copy" icon. The "Pretty" sub-tab shows the response JSON:

```
1 { 2:   "almacen": { 3:     "_id": "61132cb31a96ee2488f3ec48", 4:     "name": "store pepis", 5:     "direccion": "Av 1", 6:     "ciudad": "Cali", 7:     "__v": 0 8:   } 9: }
```

10)

The image shows the Postman application interface. At the top, there's a search bar labeled "Search Postman". Below it, a tab shows a POST request to "http://localhost:3001/api/almacen/registerAlmacen". The main area is divided into sections: "Params", "Authorization", "Headers (8)", "Body", "Pre-request Script", "Tests", and "Settings". The "Body" tab is selected, and within it, the "raw" format is chosen. The body content is a JSON object:

```
{  "name": "store time",  "direccion": "Av 5 cra 3",  "ciudad": "Bogota"}
```

. Below this, the "Body" section shows the response in "Pretty" format, displaying a JSON object:

```
{  "almacen": {    "_id": "61133f371a96ee2488f3ec4b",    "name": "store time",    "direccion": "Av 5 cra 3",    "ciudad": "Bogota",    "__v": 0  }}
```

TEST de consulta en postman

GET Producto:

The screenshot shows a Postman interface with a GET request to `http://localhost:3001/api/producto/listProducto`. The response is displayed in the 'Body' tab, showing a JSON array of product objects. The response is formatted as 'Pretty' JSON.

```
1  {
2    "producto": [
3      {
4        "_id": "611303625a904b0e6c6c8a1f",
5        "name": "Hamburguesa",
6        "precio": "10000",
7        "codigo": "111",
8        "description": "Doble carne",
9        "fecha": "2021-08-10T22:53:22.029Z",
10       "__v": 0
11     },
12     {
13       "_id": "611303af5a904b0e6c6c8a22",
14       "name": "Ajiaco",
15       "precio": "2000",
16       "codigo": "451",
17       "description": "Re caliente",
18       "fecha": "2021-08-10T22:54:39.801Z",
19       "__v": 0
20     },
21     {
22       "_id": "611304145a904b0e6c6c8a25",
23       "name": "Arepa rellena",
24       "precio": "6000",
25       "codigo": "147",
26       "description": "Super baratas",
27       "fecha": "2021-08-10T22:56:20.632Z",
28       "__v": 0
29     },
30     {
31       "_id": "611304375a904b0e6c6c8a28",
32       "name": "Salchipapas",
33       "precio": "8000",
34       "codigo": "159",
35       "description": "muchos tamaños",
36       "fecha": "2021-08-10T22:56:55.219Z",
37       "__v": 0
38     }
39   ]
40 }
```

GET Almacén

POST http://localhost:3001/api/almacen/listAlmacen [CONFLICT] POST http://localhost:3001/api/almacen/listAlmacen GET http://localhost:3001/api/almacen/listAlmacen GET http://localhost:3001/api/almacen/listAlmacen + ...

http://localhost:3001/api/almacen/listAlmacen

GET http://localhost:3001/api/almacen/listAlmacen

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
-----	-------

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "almacen": [
3     {
4       "_id": "611320d81a96ee2488f3ec39",
5       "name": "SuperEsquina",
6       "direccion": "Cra 68 No 56-45",
7       "ciudad": "Bogota",
8       "__v": 0
9     },
10    {
11      "_id": "611322501a96ee2488f3ec3c",
12      "name": "DondePepita",
13      "direccion": "Cra 67 No 26-45",
14      "ciudad": "Cali",
15      "__v": 0
16    },
17    {
18      "_id": "611322b11a96ee2488f3ec3f",
19      "name": "DondePepita Medellin",
20      "direccion": "AV 32 No 26-41",
21      "ciudad": "Medellin",
22      "__v": 0
23    },
24    {
25      "_id": "611323e61a96ee2488f3ec42",
26      "name": "La tienda Moderna",
27      "direccion": "Cra 8 av 7",
28      "ciudad": "Risaralda",
29      "__v": 0
30    },
31    {
32      "_id": "6113290b1a96ee2488f3ec45",
33      "name": "Pepite store",
```

Base de Datos

Almacén:

MongoDB Compass - localhost:27017/retodbscrumboard.almacens

Connect View Collection Help

Local

7 DBS 5 COLLECTIONS

☆ FAVORITE

HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 5.0.1 Community

Q Filter your data

- > admin
- > config
- > dbscrumboard
- > local
- > pruebadb
- > pruebadb2
- ▼ retodbscrumboard
 - almacens
 - productos

retodbscrumboard.almacens

DOCUMENTS 7 TOTAL SIZE 712B AVG. SIZE 102B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET ...

ADD DATA VIEW {}

Displaying documents 1 - 7 of 7 C REFRESH

```
{ "_id": ObjectId("611320d81a96ee2488f3ec39"),  
  "name": "SuperEsquina",  
  "direccion": "Cra 68 No 56-45",  
  "ciudad": "Bogota",  
  "__v": 0 }  
  
{ "_id": ObjectId("611322501a96ee2488f3ec3f"),  
  "name": "DondePepita",  
  "direccion": "Cra 67 No 26-45",  
  "ciudad": "Cali",  
  "__v": 0 }  
  
{ "_id": ObjectId("611322b11a96ee2488f3ec3f"),  
  "name": "DondePepita Medellin",  
  "direccion": "AV 32 No 26-41",  
  "ciudad": "Medellin",  
  "__v": 0 }  
  
{ "_id": ObjectId("611323e61a96ee2488f3ec42"),  
  "name": "La tienda Moderna",  
  "direccion": "Cra 8 av 7",  
  "ciudad": "Risaralda",  
  "__v": 0 }  
  
{ "_id": ObjectId("6113290b1a96ee2488f3ec45"),  
  "name": "DondePepita",  
  "direccion": "Cra 8 av 7",  
  "ciudad": "Risaralda",  
  "__v": 0 }
```

Productos:

MongoDB Compass - localhost:27017/retodbscrumboard.productos

Connect View Collection Help

Local

7 DBS 5 COLLECTIONS

☆ FAVORITE

HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 5.0.1 Community

Q Filter your data

- > admin
- > config
- > dbscrumboard
- > local
- > pruebadb
- > pruebadb2
- ▼ retodbscrumboard
 - almacens
 - productos

retodbscrumboard.productos

DOCUMENTS 6 TOTAL SIZE 788B AVG. SIZE 131B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET ...

ADD DATA VIEW {}

Displaying documents 1 - 6 of 6 C REFRESH

```
{ "_id": ObjectId("611303625a904b0e6cc6c8a1f"),  
  "name": "Hamburguesa",  
  "precio": "10000",  
  "codigo": "111",  
  "description": "Doble carne",  
  "fecha": "2021-08-10T22:53:22.029+00:00",  
  "__v": 0 }  
  
{ "_id": ObjectId("611303af5a904b0e6cc6c8a22"),  
  "name": "Ajiaco",  
  "precio": "2000",  
  "codigo": "451",  
  "description": "Re caliente",  
  "fecha": "2021-08-10T22:54:39.501+00:00",  
  "__v": 0 }  
  
{ "_id": ObjectId("611304145a904b0e6cc6c8a25"),  
  "name": "Arepas rellenas",  
  "precio": "6000",  
  "codigo": "147",  
  "description": "Super baratas",  
  "fecha": "2021-08-10T22:56:20.632+00:00",  
  "__v": 0 }  
  
{ "_id": ObjectId("611304375a904b0e6cc6c8a28"),  
  "name": "Salchipapas",  
  "precio": "8000",  
  "codigo": "147",  
  "description": "Super baratas",  
  "fecha": "2021-08-10T22:56:20.632+00:00",  
  "__v": 0 }
```

> _MONGOSH