

Sistema de gestión de pedidos para una tienda en línea

Como utilizar la API para los diferentes endpoints:

GET /products

The screenshot shows the Postman interface with the following details:

- Request Method:** GET
- URL:** http://127.0.0.1:8000/products
- Headers:** (4)
- Body:** ({} JSON) Preview: A JSON array of products:

```
10 |   "id": 4,
11 |   "name": "Bolígrafo",
12 |   "price": 1.5,
13 |   "category": "papelería"
14 | },
15 | {
16 |   "id": 3,
17 |   "name": "Monitor 27 Pulgadas",
18 |   "price": 150,
19 |   "category": "informática"
20 | },
21 | {
22 |   "id": 3,
23 |   "name": "Ratón inalámbrico",
24 |   "price": 20,
|   "category": "informática"
```
- Response Headers:** 200 OK, 3 ms, 399 B
- Buttons:** Save, Share, Send

GET /products/:id

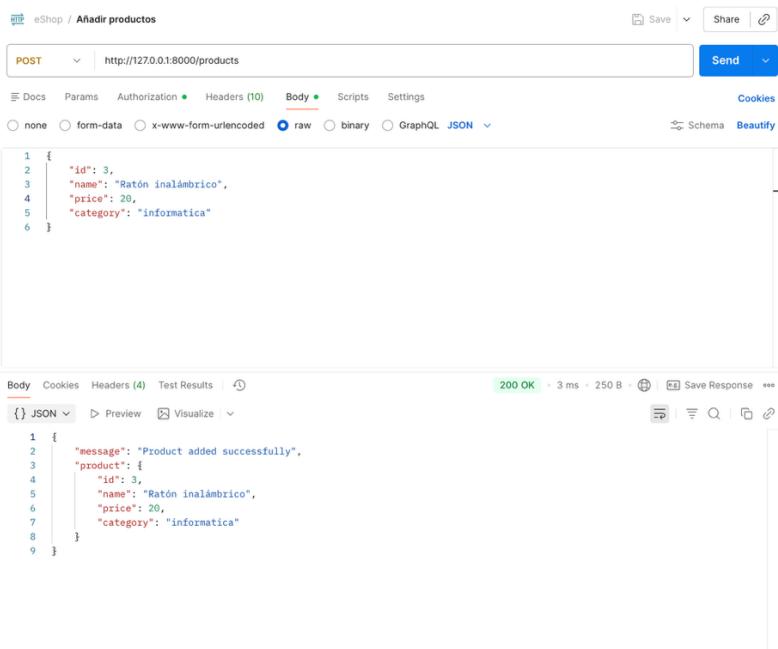
The screenshot shows the Postman interface with the following details:

- Request Method:** GET
- URL:** http://127.0.0.1:8000/products/:id
- Params:** id
- Headers:** (7)
- Body:** ({} JSON) Preview: A single product:

```
1 | {
2 |   "id": 1,
3 |   "name": "Lápiz",
4 |   "price": 0.5,
5 |   "category": "papelería"
6 | }
```
- Response Headers:** 200 OK, 5 ms, 184 B
- Buttons:** Save, Share, Send

Mario Parra Castiñeiras

POST /products



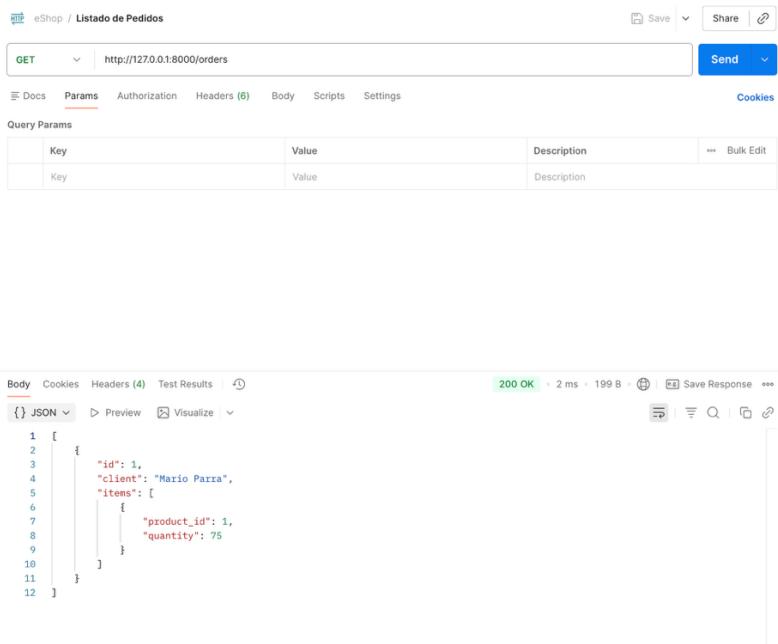
The screenshot shows a Postman request to `http://127.0.0.1:8000/products` using the `POST` method. The request body contains the following JSON:

```
1 {
2     "id": 3,
3     "name": "Ratón inalámbrico",
4     "price": 20,
5     "category": "informatica"
6 }
```

The response status is `200 OK` with a response time of 3 ms and a size of 250 B. The response body is:

```
1 {
2     "message": "Product added successfully",
3     "product": {
4         "id": 3,
5         "name": "Ratón inalámbrico",
6         "price": 20,
7         "category": "informatica"
8     }
9 }
```

GET /orders



The screenshot shows a Postman request to `http://127.0.0.1:8000/orders` using the `GET` method. The request includes the following query parameters:

Key	Value	Description
Key	Value	Description

The response status is `200 OK` with a response time of 2 ms and a size of 199 B. The response body is:

```
1 [
2     {
3         "id": 1,
4         "client": "Mario Parra",
5         "items": [
6             {
7                 "product_id": 1,
8                 "quantity": 75
9             }
10        ]
11    }
12 ]
```

Mario Parra Castiñeiras

GET /orders/:id

The screenshot shows the Postman interface with the following details:

- URL:** http://127.0.0.1:8000/orders/:id
- Method:** GET
- Params:** id = 1
- Body:** JSON response (shown below)
- Test Results:** 200 OK (3 ms, 197 B)
- Response Body (JSON):**

```
1 {  
2   "id": 1,  
3   "client": "Mario Parra",  
4   "items": [  
5     {  
6       "product_id": 1,  
7       "quantity": 75  
8     }  
9   ]  
10 }
```

POST /orders

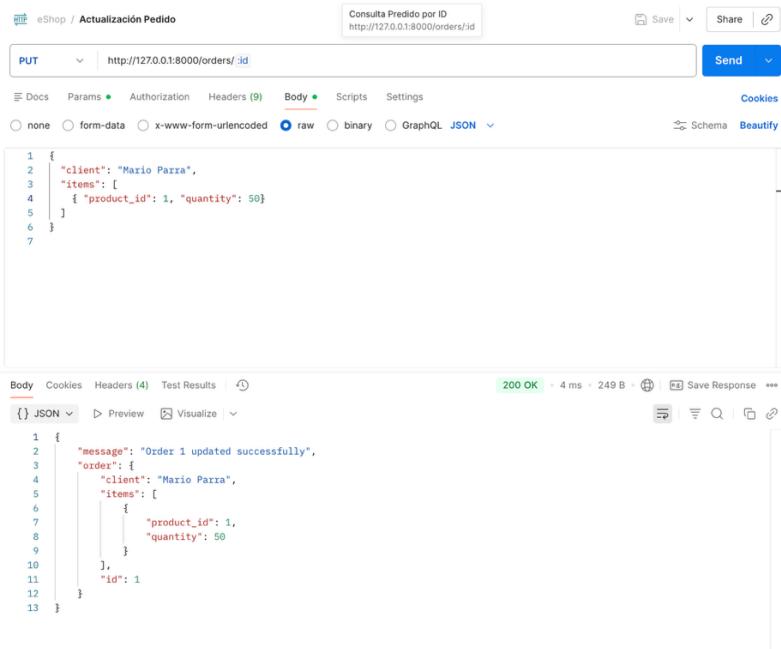
The screenshot shows the Postman interface with the following details:

- URL:** http://127.0.0.1:8000/orders
- Method:** POST
- Body:** JSON (raw)
- Test Results:** 200 OK (5 ms, 245 B)
- Response Body (JSON):**

```
1 {  
2   "message": "Order added successfully",  
3   "order": {  
4     "id": 1,  
5     "client": "Mario Parra",  
6     "items": [  
7       {  
8         "product_id": 1,  
9         "quantity": 75  
10      }  
11    ]  
12  }  
13 }
```

Mario Parra Castiñeiras

PUT /orders/:id



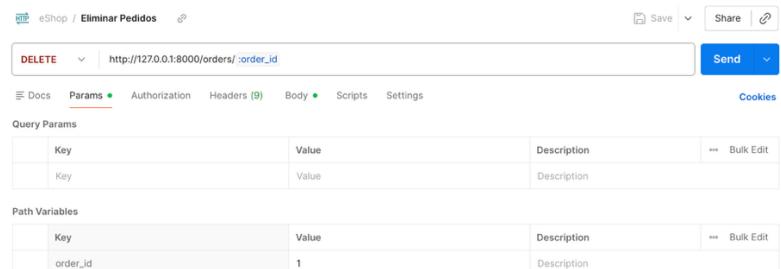
The screenshot shows a POSTMAN interface. The URL is `http://127.0.0.1:8000/orders/:id`. The method is set to `PUT`. The body contains the following JSON:

```
1 {
2   "client": "Mario Parra",
3   "items": [
4     { "product_id": 1, "quantity": 50 }
5   ]
6 }
```

The response status is `200 OK` with a response time of 4 ms and a size of 249 B. The response body is:

```
1 {
2   "message": "Order 1 updated successfully",
3   "order": {
4     "client": "Mario Parra",
5     "items": [
6       {
7         "product_id": 1,
8         "quantity": 50
9       ],
10      "id": 1
11    }
12 }
```

DELETE /orders/:id



The screenshot shows a POSTMAN interface. The URL is `http://127.0.0.1:8000/orders/:order_id`. The method is set to `DELETE`. There is a `Query Params` table with one entry: `order_id`. There is also a `Path Variables` table with one entry: `order_id` with value `1`.



The screenshot shows a POSTMAN interface. The response status is `200 OK` with a response time of 5 ms and a size of 167 B. The response body is:

```
1 {
2   "message": "Order 1 deleted successfully"
3 }
```