

# MI TIENDA ONLINE SPRINGBOOT

## TABLA CLIENTE:

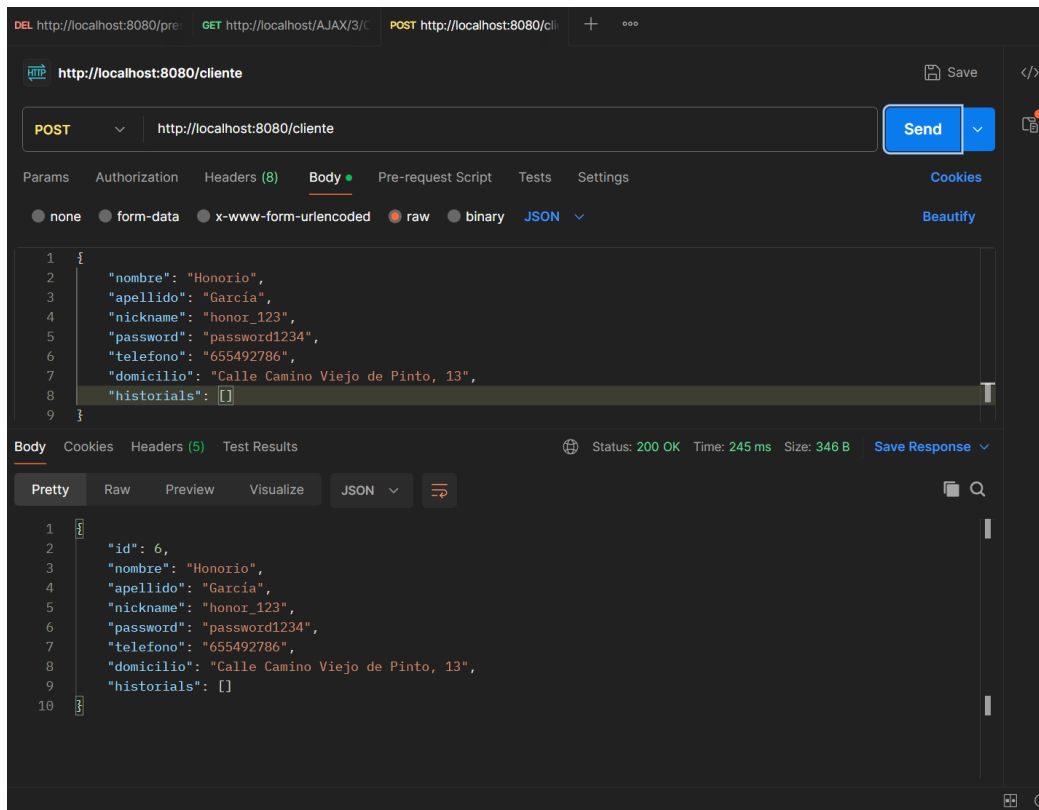
- findAll():

```
1 {
2   {
3     "id": 1,
4     "nombre": "Juan",
5     "apellido": "Pérez",
6     "nickname": "juanperez_01",
7     "password": "Clave123",
8     "telefono": "678123456",
9     "domicilio": "Calle Falsa 123, Madrid",
10    "historials": [
11      {
12        "id": 5,
13        "producto": {
14          "id": 2,
15          "nombre": "Smartphone Samsung",
16          "descripcion": "Samsung Galaxy con pantalla de 6.5 pulgadas",
17          "precio": 450.50,
18          "stock": 45,
19          "imagen_url": null
20        },
21        "fechaCompra": "2024-02-11",
```

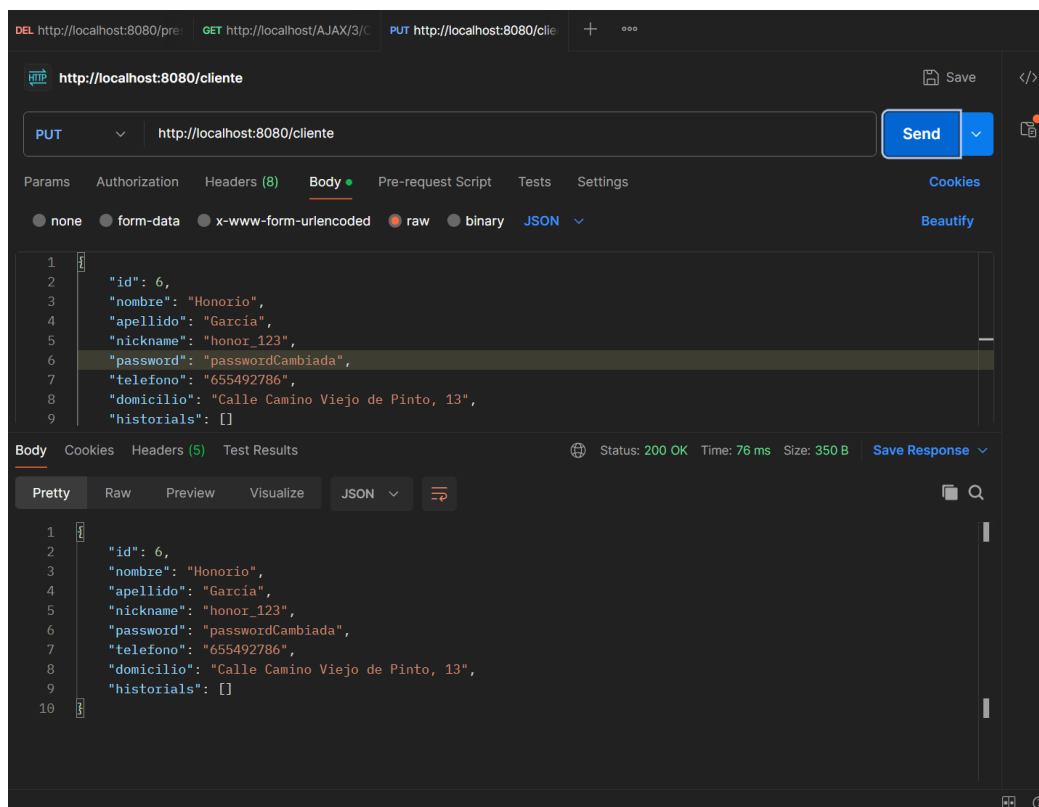
- findById():

```
1 {
2   "id": 3,
3   "nombre": "Carlos",
4   "apellido": "Ramirez",
5   "nickname": "carlos_r",
6   "password": "Segura2024",
7   "telefono": "699456789",
8   "domicilio": "Gran Via 55, Valencia",
9   "historials": [
10     {
11       "id": 3,
12       "producto": {
13         "id": 3,
14         "nombre": "Auriculares Sony",
15         "descripcion": "Auriculares inalámbricos con cancelación de ruido",
16         "precio": 120.00,
17         "stock": 15,
18         "imagen_url": null
19       },
20       "fechaCompra": "2024-02-07",
21       "cantidad": 1,
```

- insert():



- update():



- delete():

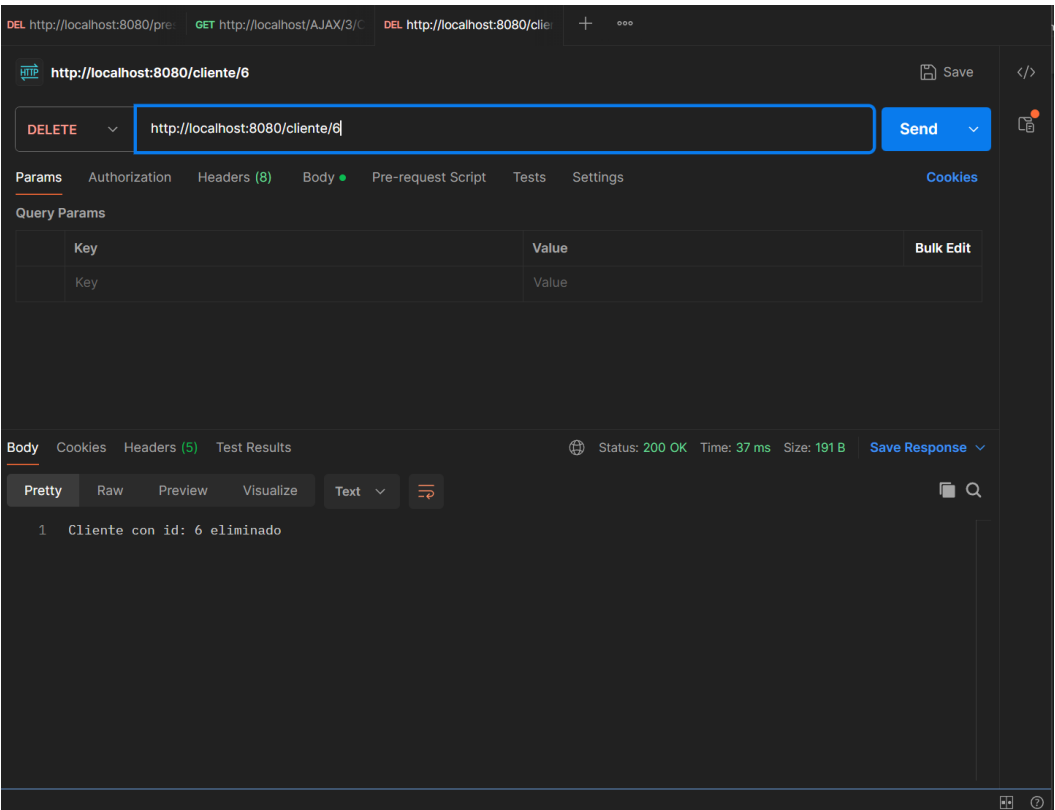
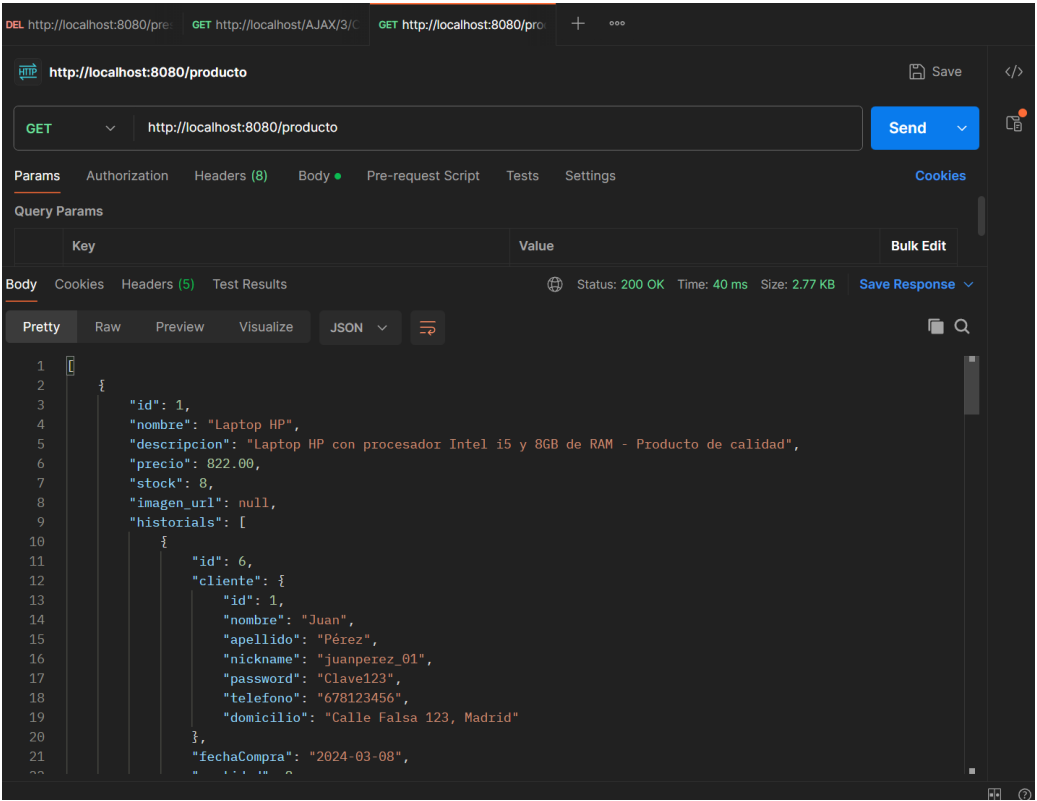


TABLA PRODUCTO:

- findAll():



- findById():

The screenshot shows a Postman interface with a GET request to `http://localhost:8080/producto/9`. The request is successful, returning a 200 OK status. The response body is a JSON object representing a product.

```
1 {
2   "id": 9,
3   "nombre": "Laptop msi",
4   "descripcion": "Laptop HP - Producto de calidad",
5   "precio": 450.00,
6   "stock": 4,
7   "imagen_url": null,
8   "historials": []
9 }
```

- insert():

The screenshot shows a Postman interface with a POST request to `http://localhost:8080/producto`. The request body is a JSON object with product details. The response is a 200 OK status, and the response body is a JSON object representing the newly inserted product.

```
1 {
2   "nombre": "Archivador",
3   "descripcion": "Tamaño A5",
4   "precio": 9.99,
5   "stock": 50,
6   "imagen_url": null,
7   "historials": []
8 }
```

- update():

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/producto`. The request body is a JSON object:

```
1 {
2   "id": 13,
3   "nombre": "Archivador Cambiado",
4   "descripcion": "Tamaño A5",
5   "precio": 9.99,
6   "stock": 50,
7   "imagen_url": null,
8   "historials": []
9 }
```

The response status is 200 OK, with a time of 19 ms and a size of 311 B. The response body is a JSON object:

```
1 {
2   "id": 13,
3   "nombre": "Archivador Cambiado",
4   "descripcion": "Tamaño A5 - Producto de oferta",
5   "precio": 9.99,
6   "stock": 50,
7   "imagen_url": null,
8   "historials": []
9 }
```

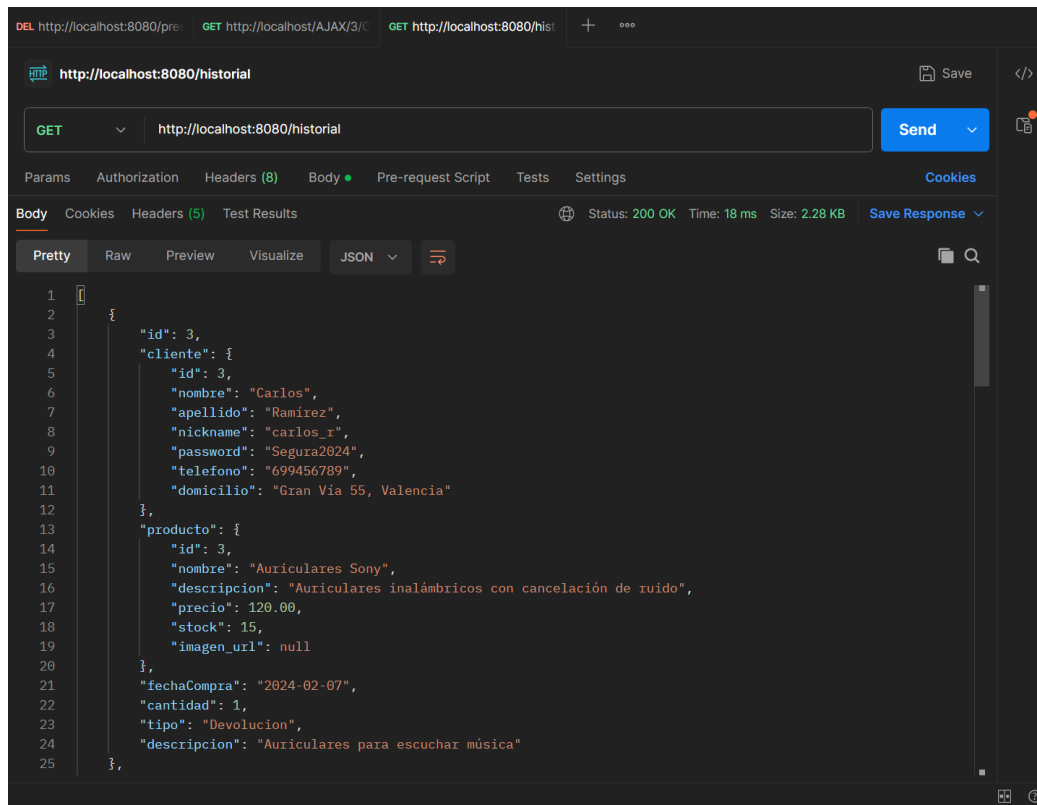
- delete():

The screenshot shows a REST client interface with a DELETE request to `http://localhost:8080/producto/13`. The response status is 200 OK, with a time of 14 ms and a size of 193 B. The response body is a text message:

```
1 Producto con id: 13 eliminado
```

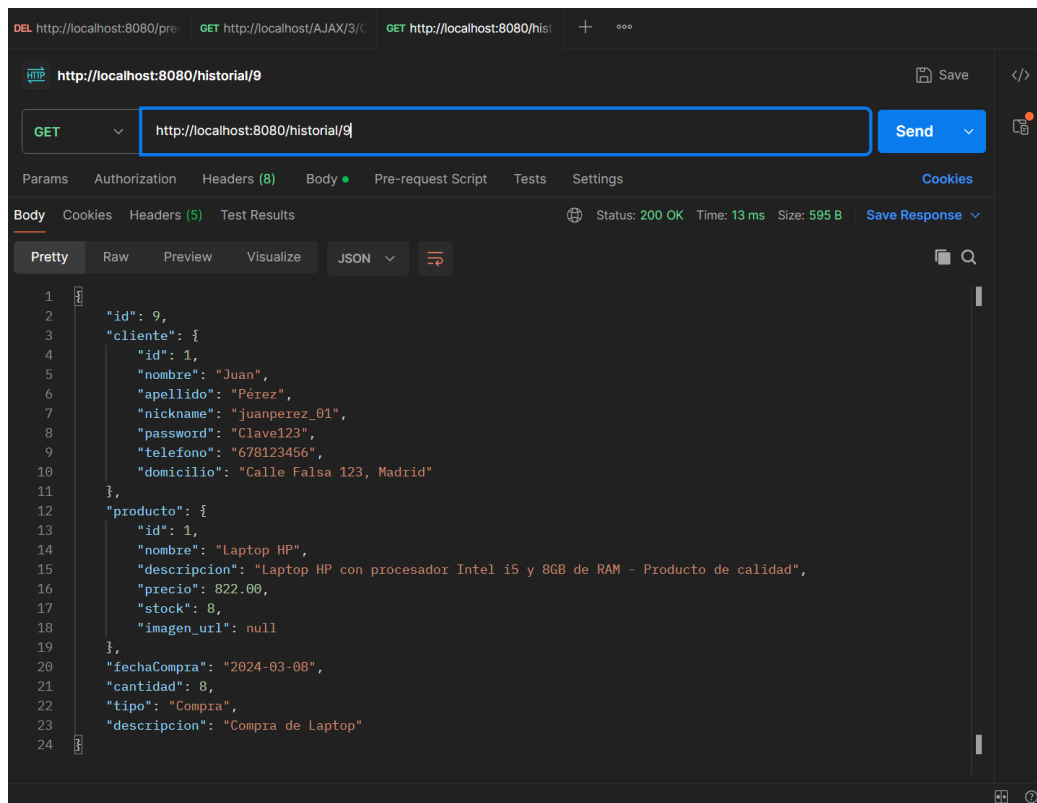
## TABLA HISTORIAL:

- findAll():



```
1 {
2   {
3     "id": 3,
4     "cliente": {
5       "id": 3,
6       "nombre": "Carlos",
7       "apellido": "Ramirez",
8       "nickname": "carlos_r",
9       "password": "Segura2024",
10      "telefono": "699456789",
11      "domicilio": "Gran Via 55, Valencia"
12    },
13    "producto": {
14      "id": 3,
15      "nombre": "Auriculares Sony",
16      "descripcion": "Auriculares inalámbricos con cancelación de ruido",
17      "precio": 120.00,
18      "stock": 15,
19      "imagen_url": null
20    },
21    "fechaCompra": "2024-02-07",
22    "cantidad": 1,
23    "tipo": "Devolucion",
24    "descripcion": "Auriculares para escuchar música"
25  },
26 }
```

- findById():



```
1 {
2   "id": 9,
3   "cliente": {
4     "id": 1,
5     "nombre": "Juan",
6     "apellido": "Pérez",
7     "nickname": "juanperez_01",
8     "password": "Clave123",
9     "telefono": "678123456",
10    "domicilio": "Calle Falsa 123, Madrid"
11  },
12  "producto": {
13    "id": 1,
14    "nombre": "Laptop HP",
15    "descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM - Producto de calidad",
16    "precio": 822.00,
17    "stock": 8,
18    "imagen_url": null
19  },
20  "fechaCompra": "2024-03-08",
21  "cantidad": 8,
22  "tipo": "Compra",
23  "descripcion": "Compra de Laptop"
24 }
```

- insert():

DEL http://localhost:8080/pre: GET http://localhost/AJAX/3/C POST http://localhost:8080/his

HTTP http://localhost:8080/historial

POST http://localhost:8080/historial

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

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "cliente": {
3     "id": 1
4   },
5   "producto": {
6     "id": 1
7   },
8   "fechaCompra": "2025-02-10",
9   "cantidad": 1,
10  "tipo": "Compra",
11  "descripcion": "Compra de Laptop"
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 176 ms Size: 596 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 13,
3   "cliente": {
4     "id": 1,
5     "nombre": "Juan",
6     "apellido": "Pérez",
7     "nickname": "juanperez_01",
8     "password": "Clave123",
9     "telefono": "678123456",
10    "domicilio": "Calle Falsa 123, Madrid"
11  },
```

- update():

DEL http://localhost:8080/pre: GET http://localhost/AJAX/3/C PUT http://localhost:8080/his

HTTP http://localhost:8080/historial

PUT http://localhost:8080/historial

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

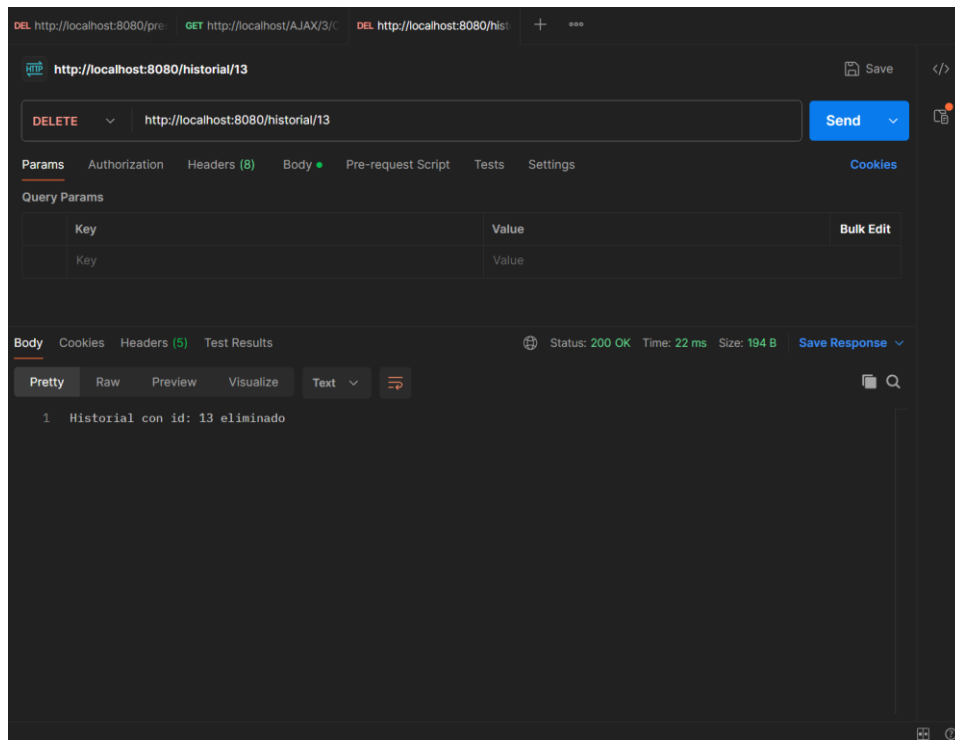
none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "id": 13,
3   "cliente": {
4     "id": 1,
5     "nombre": "Juan",
6     "apellido": "Pérez",
7     "nickname": "juanperez_01",
8     "password": "Clave123",
9     "telefono": "678123456",
10    "domicilio": "Calle Falsa 123, Madrid"
11  },
12  "producto": {
13    "id": 1,
14    "nombre": "Laptop HP",
15    "descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM - Producto de calidad",
16    "precio": 822.00,
17    "stock": 7,
18    "imagen_url": null
19  },
20  "fechaCompra": "2025-01-15",
21  "cantidad": 1,
22  "tipo": "Compra",
23  "descripcion": "Compra de Laptop Cambiada"
24 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 46 ms Size: 605 B Save Response

Pretty Raw Preview Visualize JSON

- delete():



## VALIDACIONES:

- Control del stock en compra:

```
{
  "id": 11,
  "nombre": "Laptop",
  "descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM - Producto de calidad",
  "precio": 822.00,
  "stock": 8,
  "imagen_url": null,
  "historials": []
}
```

```
{
  "producto": {
    "id": 11
  },
  "fechaCompra": "2025-01-15",
  "cantidad": 2,
  "tipo": "Compra",
  "descripcion": "Compra de Laptop Cambiada"
}
```



```
DEL http://localhost:8080/pre: GET http://localhost/AJAX/3/ GET http://localhost:8080/pro: + ***

http://localhost:8080/producto/11

GET http://localhost:8080/producto/11 Send

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

Body Cookies Headers (5) Test Results Status: 200 OK Time: 528 ms Size: 607 B Save Response

Pretty Raw Preview Visualize JSON

1 {
2   "id": 11,
3   "nombre": "Laptop",
4   "descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM - Producto de calidad",
5   "precio": 822.00,
6   "stock": 6,
7   "imagen_url": null,
8   "historials": [
9     {
10      "id": 14,
11      "cliente": {
12        "id": 1,
13        "nombre": "Juan",
14        "apellido": "Pérez",
15        "nickname": "juanperez_01",
16        "password": "Clave123",
17        "telefono": "678123456",
18        "domicilio": "Calle Falsa 123, Madrid"
19      },
20      "fechaCompra": "2025-01-15",
21      "cantidad": 2,
22      "tipo": "Compra",
23      "descripcion": "Compra de Laptop Cambiada"
24    }
25  ]
26 }
```

- En caso de que un producto no tenga existencias no puede ser comprado:

```
DEL http://localhost:8080/pre: GET http://localhost/AJAX/3/ POST http://localhost:8080/h: + ***

http://localhost:8080/historial

POST http://localhost:8080/historial Send

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

none form-data x-www-form-urlencoded raw binary JSON Beautify

1 {
2   "cliente": {
3     "id": 1
4   },
5   "producto": {
6     "id": 11
7   },
8   "fechaCompra": "2025-01-15",
9   "cantidad": 15,
10  "tipo": "Compra",
11  "descripcion": "Compra de Laptop Cambiada"
12 }

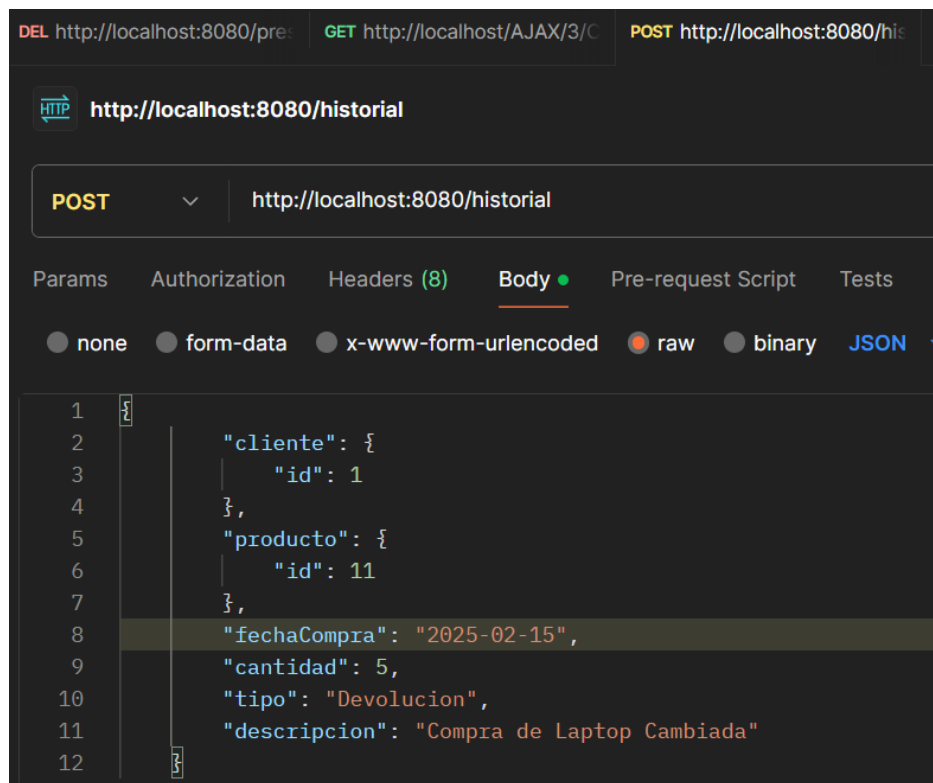
Body Cookies Headers (5) Test Results Status: 200 OK Time: 497 ms Size: 2.71 KB Save Response

Pretty Raw Preview Visualize JSON

131 {
132   "domicilio": "Calle Falsa 123, Madrid"
133 },
134 "producto": {
135   "id": 11,
136   "nombre": "Laptop",
137   "descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM - Producto de calidad",
138   "precio": 822.00,
139   "stock": 6,
140   "imagen_url": null
141 },
142 }
```

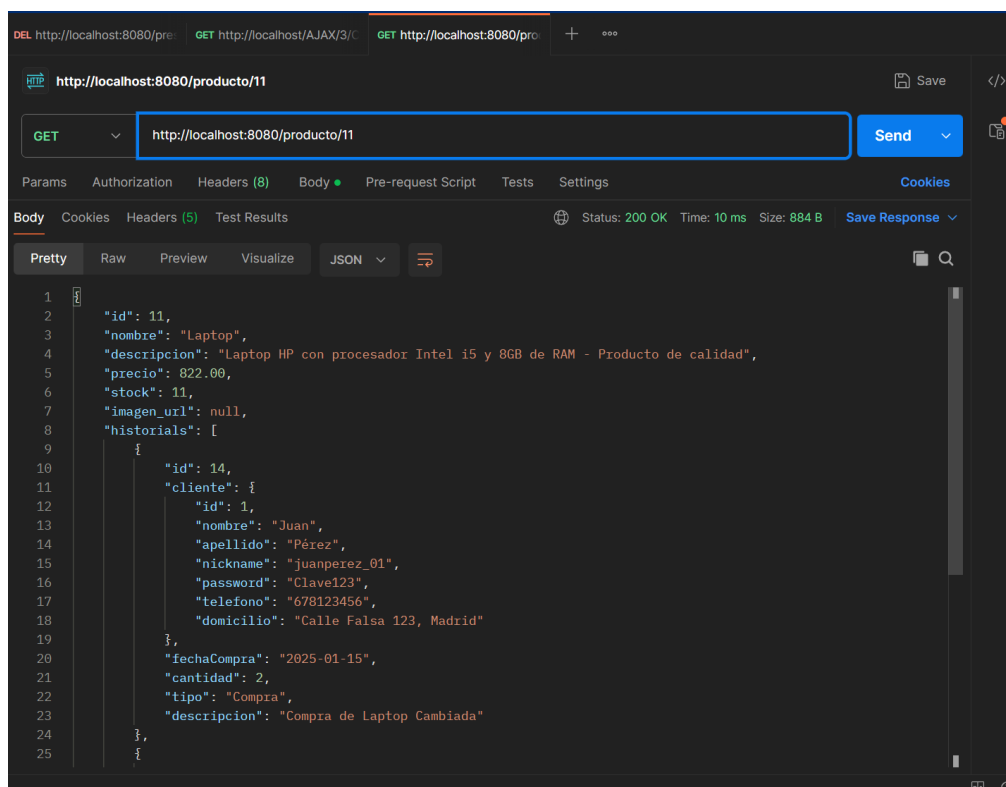
```
{
  "timestamp": "2025-02-16T13:31:14.598+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "No hay suficiente stock",
  "path": "/historial"
}
```

- Si hacemos devolución se suma el stock:



The screenshot shows a REST client interface with a POST request to `http://localhost:8080/historial`. The request body is a JSON object with the following structure:

```
1 {
2   "cliente": {
3     "id": 1
4   },
5   "producto": {
6     "id": 11
7   },
8   "fechaCompra": "2025-02-15",
9   "cantidad": 5,
10  "tipo": "Devolucion",
11  "descripcion": "Compra de Laptop Cambiada"
12 }
```

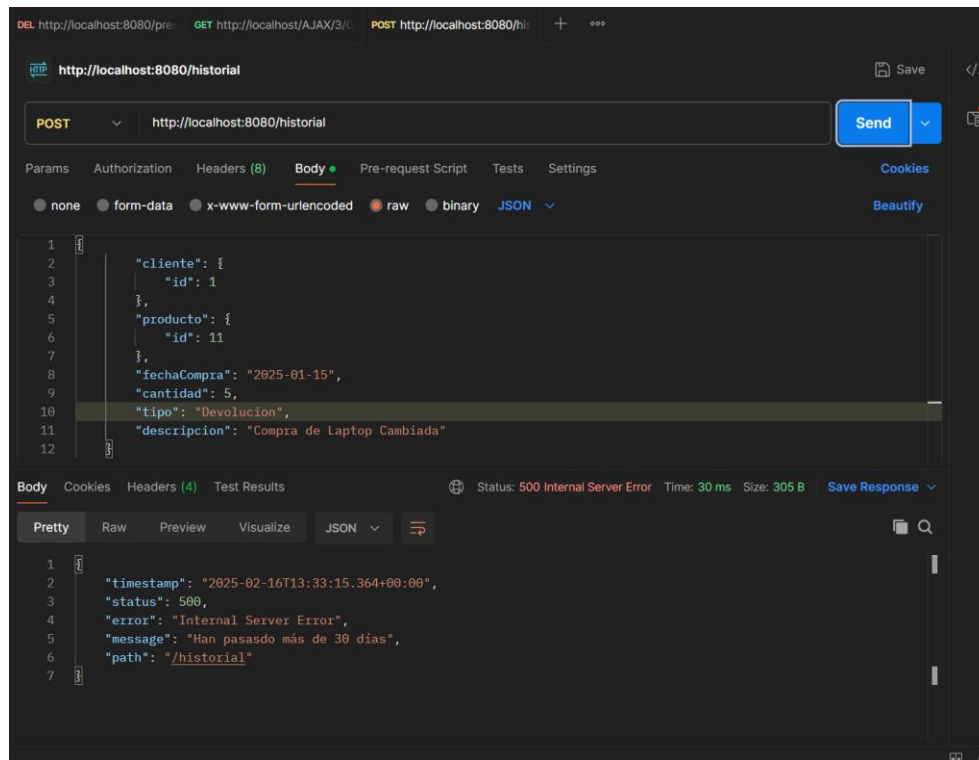


The screenshot shows a REST client interface with a GET request to `http://localhost:8080/producto/11`. The response status is 200 OK. The response body is a JSON object with the following structure:

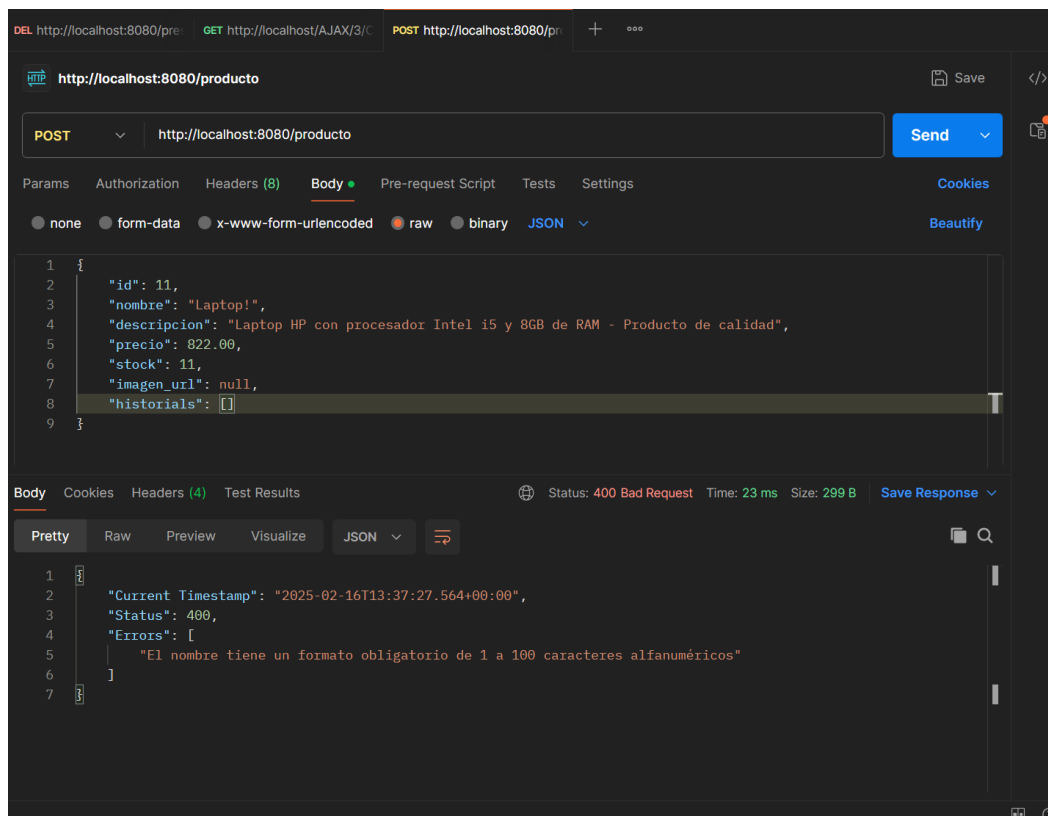
```
1 {
2   "id": 11,
3   "nombre": "Laptop",
4   "descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM - Producto de calidad",
5   "precio": 822.00,
6   "stock": 11,
7   "imagen_url": null,
8   "historials": [
9     {
10      "id": 14,
11      "cliente": {
12        "id": 1,
13        "nombre": "Juan",
14        "apellido": "Pérez",
15        "nickname": "juanperez_01",
16        "password": "Clave123",
17        "telefono": "678123456",
18        "domicilio": "Calle Falsa 123, Madrid"
19      },
20      "fechaCompra": "2025-01-15",
21      "cantidad": 2,
22      "tipo": "Compra",
23      "descripcion": "Compra de Laptop Cambiada"
24    },
25    {

```

- Para hacer una devolución no deben haber pasado más de 30 días desde a fecha de compra hasta la fecha actual:



- Nombres de productos con solo caracteres alfanuméricos:



- Nombres de clientes solo alfabéticos:

```
DEL http://localhost:8080/pre: GET http://localhost/AJAX/3/c POST http://localhost:8080/cli + ...  
  
http://localhost:8080/cliente  
  
POST http://localhost:8080/cliente  
  
Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies  
  
none form-data x-www-form-urlencoded raw binary JSON Beautify  
  
1 {  
2   "nombre": "Carlos123",  
3   "apellido": "Ramirez",  
4   "nickname": "carlos_r",  
5   "password": "Segura2024",  
6   "telefono": "699456789",  
7   "domicilio": "Gran Via 55, Valencia",  
8   "historials": []  
9 }  
  
Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 7 ms Size: 296 B Save Response  
  
Pretty Raw Preview Visualize JSON  
  
1 {  
2   "Current Timestamp": "2025-02-16T13:39:40.853+00:00",  
3   "Status": 400,  
4   "Errors": [  
5     "El nombre tiene un formato obligatorio de 1 a 50 caracteres alfabéticos"  
6   ]  
7 }
```

- Teléfono de cliente 9 dígitos que empiecen por 6 o 9:

```
DEL http://localhost:8080/pre: GET http://localhost/AJAX/3/c POST http://localhost:8080/cli + ...  
  
http://localhost:8080/cliente  
  
POST http://localhost:8080/cliente  
  
Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies  
  
none form-data x-www-form-urlencoded raw binary JSON Beautify  
  
1 {  
2   "nombre": "Carlos",  
3   "apellido": "Ramirez",  
4   "nickname": "carlos_r",  
5   "password": "Segura2024",  
6   "telefono": "5994567",  
7   "domicilio": "Gran Via 55, Valencia",  
8   "historials": []  
9 }  
  
Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 4 ms Size: 296 B Save Response  
  
Pretty Raw Preview Visualize JSON  
  
1 {  
2   "Current Timestamp": "2025-02-16T13:40:38.549+00:00",  
3   "Status": 400,  
4   "Errors": [  
5     "El teléfono tiene que empezar por 6, 7, 8 o 9 seguido de 8 dígitos más"  
6   ]  
7 }
```

- Campos obligatorios:

```
DEL http://localhost:8080/pre: GET http://localhost/AJAX/3/C POST http://localhost:8080/cli + ...  
  
http://localhost:8080/cliente  
  
POST http://localhost:8080/cliente  
  
Params Authorization Headers (8) Body Pre-request Script Tests Settings  
  
none form-data x-www-form-urlencoded raw binary JSON  
  
1 {  
2   "apellido": "Ramírez",  
3   "nickname": "carlos_r",  
4   "password": "Segura2024",  
5   "telefono": "655492791",  
6   "domicilio": "Gran Vía 55, Valencia",  
7   "historials": []  
8 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time

Pretty Raw Preview Visualize JSON

```
1 {  
2   "Current Timestamp": "2025-02-16T13:41:45.744+00:00",  
3   "Status": 400,  
4   "Errors": [  
5     "no debe ser nulo",  
6     "no debe estar vacio"  
7   ]  
8 }
```

- Contraseñas con longitud mínima, uso de mayúsculas, minúsculas y números:

```
DEL http://localhost:8080/pre: GET http://localhost/AJAX/3/C POST http://localhost:8080/cli + ...  
  
http://localhost:8080/cliente  
  
POST http://localhost:8080/cliente Send  
  
Params Authorization Headers (8) Body Pre-request Script Tests Settings  
  
none form-data x-www-form-urlencoded raw binary JSON  
  
1 {  
2   "nombre": "Ana",  
3   "apellido": "Ramírez",  
4   "nickname": "ana_",  
5   "password": "a",  
6   "telefono": "655492791",  
7   "domicilio": "Gran Vía 55, Valencia",  
8   "historials": []  
9 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 5 ms Size: 352 B Save Resp

Pretty Raw Preview Visualize JSON

```
1 {  
2   "Current Timestamp": "2025-02-16T13:44:05.416+00:00",  
3   "Status": 400,  
4   "Errors": [  
5     "La contraseña tiene que tener mínimo una letra mayúscula, una minúscula y un número y estar formada  
6       mínimo de 5 caracteres"  
7   ]  
8 }
```

- El precio de los productos debe ser un valor positivo incluyendo el 0:

DEL http://localhost:8080/pre GET http://localhost/AJAX/3/C POST http://localhost:8080/pr

HTTP http://localhost:8080/producto

POST http://localhost:8080/producto

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

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "id": 11,
3   "nombre": "Laptop",
4   "descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM - Producto de calidad",
5   "precio": -90.00,
6   "stock": 11,
7   "imagen_url": null,
8   "historials": []
9 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 7 ms Size: 255 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "Current Timestamp": "2025-02-16T13:45:19.895+00:00",
3   "Status": 400,
4   "Errors": [
5     "El precio no puede ser negativo"
6   ]
7 }
```

DEL http://localhost:8080/pre GET http://localhost/AJAX/3/C POST http://localhost:8080/pr

HTTP http://localhost:8080/producto

POST http://localhost:8080/producto

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

none form-data x-www-form-urlencoded raw binary JSON

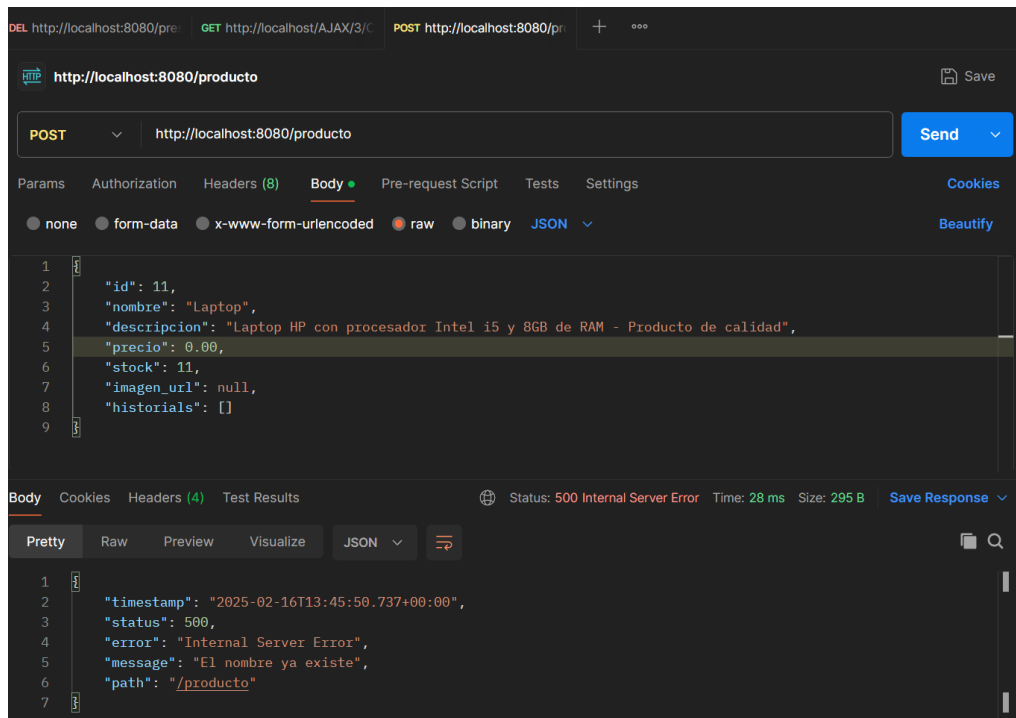
```
1 {
2   "nombre": "producto prueba",
3   "descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM",
4   "precio": 0.00,
5   "stock": 11,
6   "imagen_url": null,
7   "historials": []
8 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 9 ms Size: 343 B Save Response

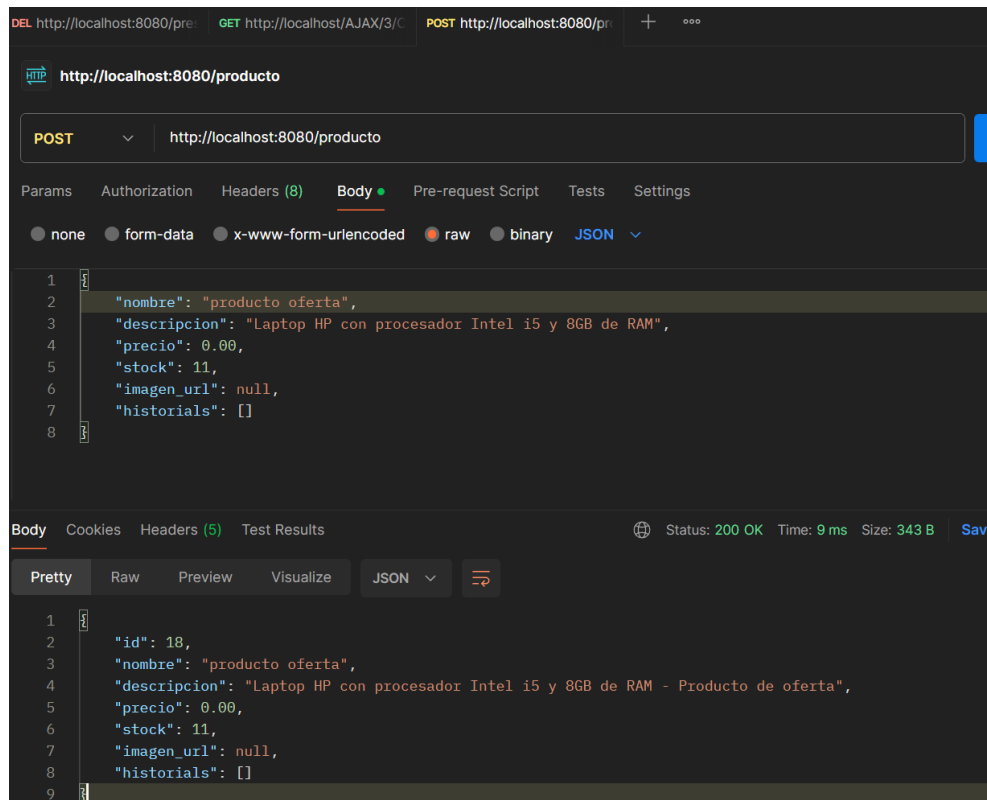
Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 17,
3   "nombre": "producto prueba",
4   "descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM - Producto de oferta",
5   "precio": 0.00,
6   "stock": 11,
7   "imagen_url": null,
8   "historials": []
9 }
```

- No se pueden repetir los nombres de los productos:



- Los productos que tengan un importe inferior a 10€ deberán tener acompañado el literal "producto de oferta". Los que tengan un precio superior a 200€ "producto de calidad":



DEL http://localhost:8080/pre

GET http://localhost/AJAX/3/C

POST http://localhost:8080/pr

+

...

HTTP

http://localhost:8080/producto

POST

▼

http://localhost:8080/producto

Params

Authorization

Headers (8)

Body ●

Pre-request Script

Tests

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

JSON ▼

1

2

3

4

5

6

7

8

9

"nombre": "producto calidad",

"descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM",

"precio": 500.00,

"stock": 11,

"imagen\_url": null,

"historials": []

Body

Cookies

Headers (5)

Test Results

⌐

Status: 200 OK

Time: 8 ms

Size: 347 B

Sa

Pretty

Raw

Preview

Visualize

JSON ▼

⌐

1

2

3

4

5

6

7

8

9

"id": 19,

"nombre": "producto calidad",

"descripcion": "Laptop HP con procesador Intel i5 y 8GB de RAM - Producto de calidad",

"precio": 500.00,

"stock": 11,

"imagen\_url": null,

"historials": []