

## VALIDACIÓN DE ENDPOINTS Y CÓDIGOS DE ESTADO HTTP

Mediante pruebas funcionales con Postman

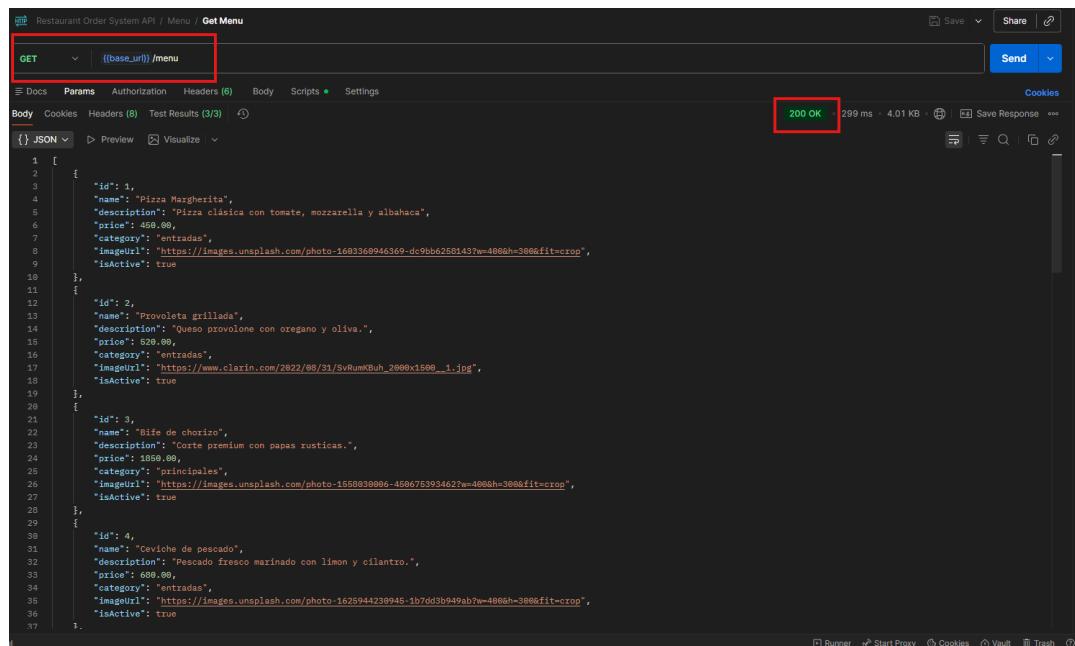
Aseguramiento de la Calidad / QA / Desarrollo Web  
Training Sofka

Para la ejecución de las pruebas funcionales, la URL base de la API y el token debe configurarse utilizando **Environments de Postman**

Medellín, Colombia  
2026

## Menú:

Get: <http://localhost:8080/menu>



The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** `(base_url) /menu`
- Response Status:** 200 OK
- Response Time:** 299 ms
- Response Size:** 4.01 KB
- Headers:** (8)
- Body (JSON):** A JSON array of menu items. Each item has fields: id, name, description, price, category, imageUrl, and isActive. The first item is "Pizza Margherita".

```
[{"id": 1, "name": "Pizza Margherita", "description": "Pizza cl\u00e1sica con tomate, mozzarella y albahaca.", "price": 468.00, "category": "entradas", "imageUrl": "https://images.unsplash.com/photo-1683369946369-dc9bb6258143?w=488&h=388&fit=crop", "isActive": true}, {"id": 2, "name": "Provoleta grillada", "description": "Queso provolone con or\u00f3geno y oliva.", "price": 620.00, "category": "entradas", "imageUrl": "https://www.cisarin.com/2022/08/31/SvRnMQuN_2000x1500_1.jpg", "isActive": true}, {"id": 3, "name": "Bife de chorizo", "description": "Corte premium con papas r\u00fasticas.", "price": 1098.00, "category": "principales", "imageUrl": "https://images.unsplash.com/photo-1556939086-450675395462?w=488&h=388&fit=crop", "isActive": true}, {"id": 4, "name": "Ceviche de pescado", "description": "Pescado fresco marinado con lim\u00f3n y cilantro.", "price": 698.00, "category": "entradas", "imageUrl": "https://images.unsplash.com/photo-1628944238945-1b7dd3b949ab?w=488&h=388&fit=crop", "isActive": true}]
```

## Crear una orden

POST: <http://localhost:8080/orders>

Content-Type: application/json

```
{
  "tableId": "1",
  "items": [
    {
      "productId": "2",
      "quantity": "2",
      "note": "sin piña"
    },
    {
      "productId": "5",
      "quantity": "2"
    }
  ]
}
```

The screenshot shows the Postman interface for a POST request to the '/orders' endpoint. The URL in the header is {{base\_url}}/orders. The response status is 201 Created, indicating success. The request body is a JSON object representing an order with items and their details.

```
1 {
2   "id": "221b69f7-eed4-4a67-b857-966033878995",
3   "tableId": 1,
4   "status": "PENDING",
5   "items": [
6     {
7       "id": 7,
8       "productId": 2,
9       "productName": "Provoleta grillada",
10      "quantity": 2,
11      "note": "sin pina"
12    },
13    {
14      "id": 8,
15      "productId": 5,
16      "productName": "Tabla de fiambres",
17      "quantity": 2,
18      "note": null
19    }
20  ],
21  "createdAt": "2026-02-26T02:44:35.489983735",
22  "updatedAt": "2026-02-26T02:44:35.489998364"
23 }
```

## Get Order by Id

**GET:** {{base\_url}}/orders/221b69f7-eed4-4a67-b857-966033878995

The screenshot shows the Postman interface for a GET request to the '/orders/221b69f7-eed4-4a67-b857-966033878995' endpoint. The URL in the header is {{base\_url}}/orders/221b69f7-eed4-4a67-b857-966033878995. The response status is 200 OK, indicating success. The response body is identical to the one shown in the previous screenshot.

```
1 {
2   "id": "221b69f7-eed4-4a67-b857-966033878995",
3   "tableId": 1,
4   "status": "PENDING",
5   "items": [
6     {
7       "id": 7,
8       "productId": 2,
9       "productName": "Provoleta grillada",
10      "quantity": 2,
11      "note": "sin pina"
12    },
13    {
14      "id": 8,
15      "productId": 5,
16      "productName": "Tabla de fiambres",
17      "quantity": 2,
18      "note": null
19    }
20  ],
21  "createdAt": "2026-02-26T02:44:35.489983735",
22  "updatedAt": "2026-02-26T02:44:35.489998364"
23 }
```

## get all orders

**GET:** {{base\_url}}/orders

## get orders by status

**GET:** {{base\_url}}/orders?status=PENDING

The screenshot shows a POSTMAN interface with the following details:

- Method:** GET
- URL:** {{base\_url}} /orders?status=PENDING
- Headers:** Authorization, Headers (7), Body, Scripts, Settings
- Query Params:** status = PENDING
- Body:** Cookies, Headers (8), Test Results (2/2)
- Status:** 200 OK (highlighted with a red box)
- Response Preview:** JSON response showing an order with items and their details.

```
{} JSON ▾ ▷ Preview ▷ Visualize ▾
1 [ {
2   "id": "a6ec1ceb-51a1-4f2b-adb3-f5342d2f154e",
3   "customerId": 1,
4   "status": "PENDING",
5   "items": [
6     {
7       "id": 3,
8       "productId": 1,
9       "productName": "Pizza Margherita",
10      "quantity": 2,
11      "note": "Sin cebolla"
12    },
13    {
14      "id": 4,
15      "productId": 5,
16      "productName": "Tabla de fiambres",
17      "quantity": 1,
18      "note": null
19    }
20  ],
21  "createdAt": "2026-02-26T02:18:45.843Z",
22  "updatedAt": "2026-02-26T02:18:45.843Z"
23 },
24 {
25   "id": "68a54c37-fe45-41bb-a7a8-e67710ee029e",
26   "customerId": 5,
27   "status": "PENDING",
28   "items": [
29     {
30       "id": 1,
31       "productId": 2,
32       "productName": "Sándwich de jamón y queso",
33       "quantity": 1,
34       "note": "Con lechuga y tomate"
35     }
36   ],
37   "createdAt": "2026-02-26T02:18:45.843Z",
38   "updatedAt": "2026-02-26T02:18:45.843Z"
39 } ]
```

## Update order status

PATCH: {{base\_url}}/orders/221b69f7-eed4-4a67-b857-966033878995/status

The screenshot shows a Postman request for a PATCH operation. The URL is {{base\_url}}/orders/221b69f7-eed4-4a67-b857-966033878995/status. The Body tab is selected, showing raw JSON with a single field "status": "READY". The response status is 200 OK.

```
1 {
2   "id": "221b69f7-eed4-4a67-b857-966033878995",
3   "status": "READY",
4   "items": [
5     {
6       "id": 7,
7       "productId": 2,
8       "productName": "Provoleta grillada",
9       "quantity": 2,
10      "note": "sin pina"
11    },
12    {
13      "id": 8,
14      "productId": 6,
15      "productName": "Tabla de fiambres",
16      "quantity": 2,
17      "note": null
18    }
19  ],
20  "createdAt": "2026-02-26T02:44:35.489984",
21  "updatedAt": "2026-02-26T02:55:33.191096063"
22 }
```

## Delete order by Id

Delete: {{base\_url}}/orders/f5a54c37-fe45-41bb-a7a8-e67710ee020e

The screenshot shows a Postman request for a DELETE operation. The URL is {{base\_url}}/orders/f5a54c37-fe45-41bb-a7a8-e67710ee020e. The Body tab is selected, showing raw JSON with a single digit "1". The response status is 204 No Content.

## Error cases

### Create order -invalid TableId

POST: {{base\_url}}/orders

```
{
  "tableId": "YOUR_INVALID_TABLE_ID",
  "items": [
    {
      "productId": "YOUR_PRODUCT_ID",
      "quantity": 2
    }
  ]
}
```

The screenshot shows the Postman interface. The request method is set to POST, and the URL is {{base\_url}}/orders. The body contains a JSON object with a 'tableId' key set to 'YOUR\_INVALID\_TABLE\_ID'. The response status is 400 Bad Request, with a timestamp of 2026-02-26T15:17:24.624314831, status 400, error 'Bad Request', and message 'Malformed request body or invalid value'.

```
POST {{base_url}}/orders
{
  "tableId": "YOUR_INVALID_TABLE_ID",
  "items": [
    {
      "productId": "YOUR_PRODUCT_ID",
      "quantity": 2
    }
  ]
}
```

400 Bad Request · 30 ms · 365 B · Save Response

## create order - product not found

**POST:** {{base\_url}}/orders

```
{
  "tableId": "YOUR_TABLE_ID",
  "items": [
    {
      "productId": "YOUR_PRODUCT_ID",
      "quantity": "YOUR_QUANTITY"
    }
  ]
}
```

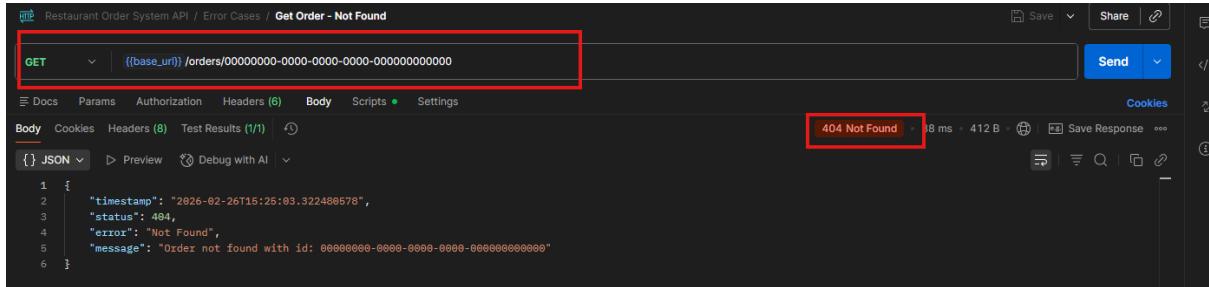
The screenshot shows the Postman interface. The request method is set to POST, and the URL is {{base\_url}}/orders. The body contains a JSON object with a 'tableId' key set to 'YOUR\_TABLE\_ID'. The response status is 400 Bad Request, with a timestamp of 2026-02-26T15:31:48.462007425, status 400, error 'Bad Request', and message 'Malformed request body or invalid value'.

```
POST {{base_url}}/orders
{
  "tableId": "YOUR_TABLE_ID"
}
```

400 Bad Request · 10 ms · 365 B · Save Response

## get order - not found

GET: {{base\_url}}/orders/00000000-0000-0000-0000-000000000000



Restaurant Order System API / Error Cases / Get Order - Not Found

GET | {{base\_url}} /orders/00000000-0000-0000-0000-000000000000

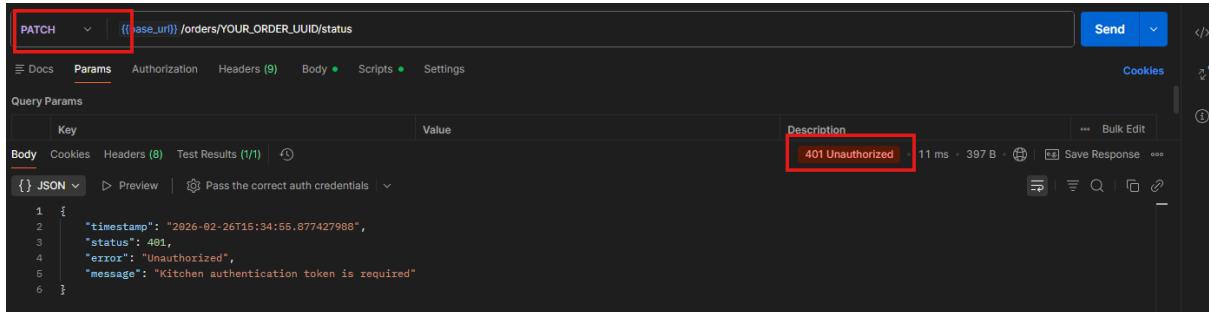
Body | Headers (8) | Body | Scripts | Settings | Cookies

404 Not Found | 18 ms | 412 B | Save Response

```
1 {  
2   "timestamp": "2026-02-26T15:25:03.322486578",  
3   "status": 404,  
4   "error": "Not Found",  
5   "message": "Order not found with id: 00000000-0000-0000-0000-000000000000"  
6 }
```

## Updated status - no token

PATCH: {{base\_url}}/orders/6f2fa0c0-f7b4-4834-8388-7261cf719ddb/status



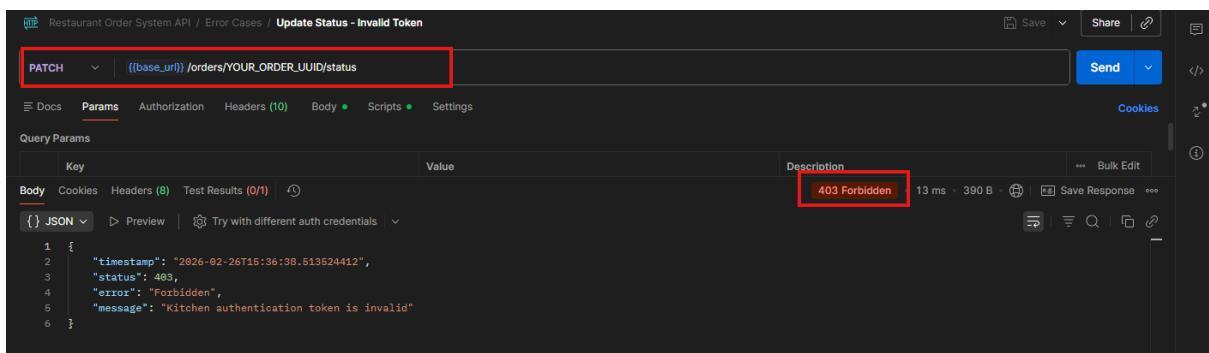
PATCH | {{base\_url}} /orders/YOUR\_ORDER\_UUID/status

Body | Headers (8) | Test Results (1/1) | Description | 401 Unauthorized | 11 ms | 397 B | Save Response

```
1 {  
2   "timestamp": "2026-02-26T15:34:55.877427988",  
3   "status": 401,  
4   "error": "Unauthorized",  
5   "message": "Kitchen authentication token is required"  
6 }
```

## Updated status - invalid token

PATCH: {{base\_url}}/orders/6f2fa0c0-f7b4-4834-8388-7261cf719ddb/status



PATCH | {{base\_url}} /orders/YOUR\_ORDER\_UUID/status

Body | Headers (8) | Test Results (0/1) | Description | 403 Forbidden | 13 ms | 390 B | Save Response

```
1 {  
2   "timestamp": "2026-02-26T15:36:38.513524412",  
3   "status": 403,  
4   "error": "Forbidden",  
5   "message": "Kitchen authentication token is invalid"  
6 }
```

## Update status - invalid status value

**PATCH:** {{base\_url}}/orders/6f2fa0c0-f7b4-4834-8388-7261cf719ddb/status

The screenshot shows the Postman interface for a PATCH request to update an order status. The URL is set to {{base\_url}}/orders/6f2fa0c0-f7b4-4834-8388-7261cf719ddb/status. The request body contains a single key-value pair: "status": "INVALID\_STATUS". The response status is 400 Bad Request, with a timestamp of 2026-02-26T15:37:50.914Z, status 400, error "Bad Request", and message "Malformed request body or invalid value".

```
1 {
2   "status": "INVALID_STATUS"
3 }
```

400 Bad Request · 13 ms · 365 B · Save Response

```
1 {
2   "timestamp": "2026-02-26T15:37:50.914Z",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Malformed request body or invalid value"
6 }
```

## Reports

### Get report By Date Range

GET: {{report\_url}}/reports?startDate=2026-02-27&endDate=2026-02-27

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** {{report\_url}}/reports?startDate=2026-02-27&endDate=2026-02-27
- Headers:** (6)
- Body:** (empty)
- Query Params:**

Key	Value	Description
startDate	2026-02-27	
endDate	2026-02-27	
- Response:** 200 OK | 21 ms | 386 B | Save Response
- Preview:** JSON (shown below)

```
1 {  
2   "totalReadyOrders": 1,  
3   "totalRevenue": 0,  
4   "productBreakdown": [  
5     {  
6       "productId": 1,  
7       "productName": null,  
8       "quantitySold": 2,  
9       "totalAccumulated": 0  
10      }  
11    ]  
12 }
```

## Get Report - Invalid Date Format

GET: {{report\_url}}/reports?startDate=26-02-28&endDate=2026-02-28

The screenshot shows a REST API testing interface with the following details:

- Method:** GET (highlighted with a red box)
- URL:** {{report\_url}} /reports?startDate=26-02-28&endDate=2026-02-28
- Headers:** (6 items)
- Body:** (Empty)
- Query Params:**

Key	Value	Description	Bulk Edit
startDate	26-02-28		
endDate	2026-02-28		
- Cookies:** (Empty)

**Response:**

- Status:** 400 Bad Request (highlighted with a red box)
- Timestamp:** 2026-02-27T19:25:12.202712125
- Status:** 400
- Error:** Bad Request
- Message:** Invalid date format: 26-02-28