

## CRUD tabla usuarios

GET <http://localhost:8080/api/usuarios/>

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

- Method:** GET
- URL:** <http://localhost:8080/api/usuarios/>
- Status:** 200 OK
- Time:** 455 ms
- Size:** 729 B

The response body is a JSON array of three user objects:

```
[
  {
    "idUsuario": 1,
    "nombreUsuario": "UsuarioUno",
    "password": "password123",
    "email": "usuario1@example.com",
    "fechaCreacion": "2025-03-07T12:00:00"
  },
  {
    "idUsuario": 2,
    "nombreUsuario": "UsuarioDos",
    "password": "securePass456",
    "email": "usuario2@example.com",
    "fechaCreacion": "2025-03-07T12:30:00"
  },
  {
    "idUsuario": 3,
    "nombreUsuario": "UsuarioTres",
    "password": "anotherPass789",
    "email": "usuario3@example.com",
    "fechaCreacion": "2025-03-07T13:00:00"
  }
]
```

GET <http://localhost:8080/api/usuarios/2>

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

- Method:** GET
- URL:** <http://localhost:8080/api/usuarios/2>
- Status:** 200 OK
- Time:** 47 ms
- Size:** 304 B

The response body is a JSON object representing the second user:

```
{
  "idUsuario": 2,
  "nombreUsuario": "UsuarioDos",
  "password": "securePass456",
  "email": "usuario2@example.com",
  "fechaCreacion": "2025-03-07T12:30:00"
}
```

POST <http://localhost:8080/api/usuarios/>

POST <http://localhost:8080/api/usuarios/> Send

Params Authorization Headers (9) **Body** Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombreUsuario": "UsurioCreado",
3   "email": "creado@example.com",
4   "password": "kjpgukdag7"
5 }
```

Body Cookies Headers (5) Test Results 201 Created • 194 ms • 313 B Save Response

{ JSON Preview Visualize

```
1 {
2   "idUsuario": 12,
3   "nombreUsuario": "UsurioCreado",
4   "password": "kjpgukdag7",
5   "email": "creado@example.com",
6   "fechaCreacion": "2025-04-10T16:37:18.676817"
7 }
```

PUT <http://localhost:8080/api/usuarios/4>

PUT <http://localhost:8080/api/usuarios/4> Send

Params Authorization Headers (9) **Body** Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombreUsuario": "Jesus Nolasco",
3   "email": "nolasco@example.com",
4   "password": "dhhihq78"
5 }
```

Body Cookies Headers (5) Test Results 200 OK • 16 ms • 307 B Save Response

{ JSON Preview Visualize

```
1 {
2   "idUsuario": 4,
3   "nombreUsuario": "Jesus Nolasco",
4   "password": "dhhihq78",
5   "email": "nolasco@example.com",
6   "fechaCreacion": "2025-04-09T19:26:15.45795"
7 }
```

DELETE <http://localhost:8080/api/usuarios/12>

The screenshot shows a REST client interface with a DELETE request to `http://localhost:8080/api/usuarios/12`. The interface includes tabs for Params, Authorization, Headers (9), Body, Scripts, Tests, and Settings. The response status is **204 No Content** with a response time of 29 ms and a size of 112 B. The response body is empty.

CRUD tabla categorías

GET <http://localhost:8080/api/categoria/>

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/categoria/`. The response status is **200 OK** with a response time of 33 ms and a size of 681 B. The response body is a JSON array of category objects.

```
1 [
2   {
3     "idCategoria": 4,
4     "descripcion": "Categoría de tecnología y gadgets",
5     "nombre": "Tecnología"
6   },
7   {
8     "idCategoria": 5,
9     "descripcion": "Categoría de libros y literatura",
10    "nombre": "Libros"
11  },
12  {
13    "idCategoria": 6,
14    "descripcion": "Categoría de ropa y accesorios",
15    "nombre": "Moda"
16  },
17  {
```

GET <http://localhost:8080/api/categoria/4>

GET <http://localhost:8080/api/categoria/4> Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results 200 OK • 11 ms • 256 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "idCategoria": 4,
3   "descripcion": "Categoría de tecnología y gadgets",
4   "nombre": "Tecnología"
5 }
```

POST <http://localhost:8080/api/categoria/>

POST <http://localhost:8080/api/categoria/> Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombre": "Electrónica",
3   "descripcion": "Dispositivos electrónicos y accesorios"
4 }
5
```

Body Cookies Headers (5) Test Results 201 Created • 20 ms • 267 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "idCategoria": 11,
3   "descripcion": "Dispositivos electrónicos y accesorios",
4   "nombre": "Electrónica"
5 }
```

PUT <http://localhost:8080/api/categoria/10>

PUT <http://localhost:8080/api/categoria/10> Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombre": "Cosmeticos",
3   "descripcion": "Accesorios de belleza"
4 }
5
```

Body Cookies Headers (5) Test Results 200 OK • 15 ms • 242 B • Save Response

JSON Preview Visualize

```
1 {
2   "idCategoria": 10,
3   "descripcion": "Accesorios de belleza",
4   "nombre": "Cosmeticos"
5 }
```

DELETE <http://localhost:8080/api/categoria/10>

DELETE <http://localhost:8080/api/categoria/10> Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1
```

Body Cookies Headers (3) Test Results 204 No Content • 16 ms • 112 B • Save Response

Raw Preview Visualize

```
1
```

## CRUD tabla productos

GET <http://localhost:8080/api/productos/>

GET <http://localhost:8080/api/productos/> Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results 200 OK • 62 ms • 1.66 KB Save Response

{ } JSON Preview Visualize

```
1 [
2   {
3     "idProducto": 10,
4     "nombreProducto": "Celular Samsung Galaxy S27",
5     "descripcion": "Última generación",
6     "precio": 25000.00,
7     "stock": 20,
8     "imagen": "galaxy_s26.jpg",
9     "fechaCreacion": "2025-03-07T10:00:00",
10    "idCategoria": 4
11  },
12  {
13    "idProducto": 11,
14    "nombreProducto": "Laptop Ultrabook",
15    "descripcion": "Laptop ultradelgada con procesador de última generación",
16    "precio": 1299.99,
17    "stock": 30,
```

GET <http://localhost:8080/api/productos/10>

GET <http://localhost:8080/api/productos/10> Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results 200 OK • 12 ms • 372 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "idProducto": 10,
3   "nombreProducto": "Celular Samsung Galaxy S27",
4   "descripcion": "Última generación",
5   "precio": 25000.00,
6   "stock": 20,
7   "imagen": "galaxy_s26.jpg",
8   "fechaCreacion": "2025-03-07T10:00:00",
9   "idCategoria": 4
10 }
```

POST <http://localhost:8080/api/productos/>

POST <http://localhost:8080/api/productos/> Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombreProducto": "Celular Samsung",
3   "descripcion": "S25 ultra",
4   "precio": 23000.50,
5   "stock": 15,
6   "imagen": "s25ultra.jpg",
7   "idCategoria": 4
8 }
```

Body Cookies Headers (5) Test Results 201 Created • 56 ms • 363 B Save Response

{ JSON Preview Visualize

```
1 {
2   "idProducto": 17,
3   "nombreProducto": "Celular Samsung",
4   "descripcion": "S25 ultra",
5   "precio": 23000.50,
6   "stock": 15,
7   "imagen": "s25ultra.jpg",
8   "fechaCreacion": "2025-04-10T16:45:31.0944705",
9   "idCategoria": 4
10 }
```

PUT <http://localhost:8080/api/productos/10>

PUT <http://localhost:8080/api/productos/10> Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombreProducto": "Celular Samsung Galaxy S27",
3   "descripcion": "Última generación",
4   "precio": 25000.00,
5   "stock": 20,
6   "imagen": "galaxy_s26.jpg",
7   "idCategoria": 4
8 }
```

Body Cookies Headers (5) Test Results 200 OK • 51 ms • 372 B Save Response

{ JSON Preview Visualize

```
1 {
2   "idProducto": 10,
3   "nombreProducto": "Celular Samsung Galaxy S27",
4   "descripcion": "Última generación",
5   "precio": 25000.00,
6   "stock": 20,
7   "imagen": "galaxy_s26.jpg",
8   "fechaCreacion": "2025-03-07T10:00:00",
9   "idCategoria": 4
10 }
```

DELETE <http://localhost:8080/api/productos/10>

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

- Method:** DELETE
- URL:** <http://localhost:8080/api/productos/10>
- Send Button:** Send
- Params:** Authorization, Headers (7), Body, Scripts, Tests, Settings
- Query Params:** A table with columns Key, Value, and Description. It contains one row with the value 'Key'.
- Body:** Cookies, Headers (3), Test Results
- Response:** 204 No Content, 13 ms, 112 B
- Raw View:** A dropdown menu showing 'Raw' selected.
- Preview View:** A dropdown menu showing 'Preview' selected.
- Visualize View:** A dropdown menu showing 'Visualize' selected.

CRUD tabla pedidos

GET <http://localhost:8080/api/pedido/>

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

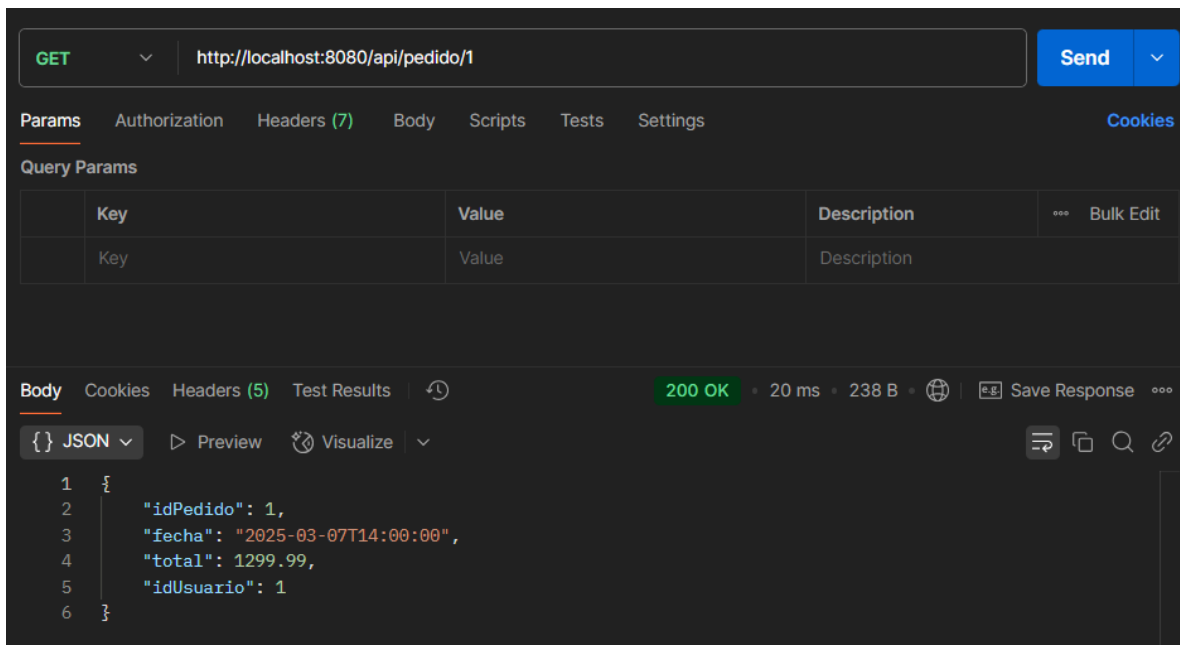
- Method:** GET
- URL:** <http://localhost:8080/api/pedido/>
- Send Button:** Send
- Params:** Authorization, Headers (7), Body, Scripts, Tests, Settings
- Query Params:** A table with columns Key, Value, and Description. It contains one row with the value 'Key'.
- Body:** Cookies, Headers (5), Test Results
- Response:** 200 OK, 36 ms, 460 B
- JSON View:** A dropdown menu showing 'JSON' selected.
- Preview View:** A dropdown menu showing 'Preview' selected.
- Visualize View:** A dropdown menu showing 'Visualize' selected.

The JSON response is as follows:

```
1 [
2   {
3     "idPedido": 1,
4     "fecha": "2025-03-07T14:00:00",
5     "total": 1299.99,
6     "idUsuario": 1
7   },
8   {
9     "idPedido": 2,
10    "fecha": "2025-03-08T10:00:00",
11    "total": 150.0,
12    "idUsuario": 2
13  },
14  {
15    "idPedido": 3,
16    "fecha": "2025-03-09T15:30:00",
17    "total": 999.99,
18    "idUsuario": 1
19  }
20 ]
```



GET <http://localhost:8080/api/pedido/1>



GET <http://localhost:8080/api/pedido/1> Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

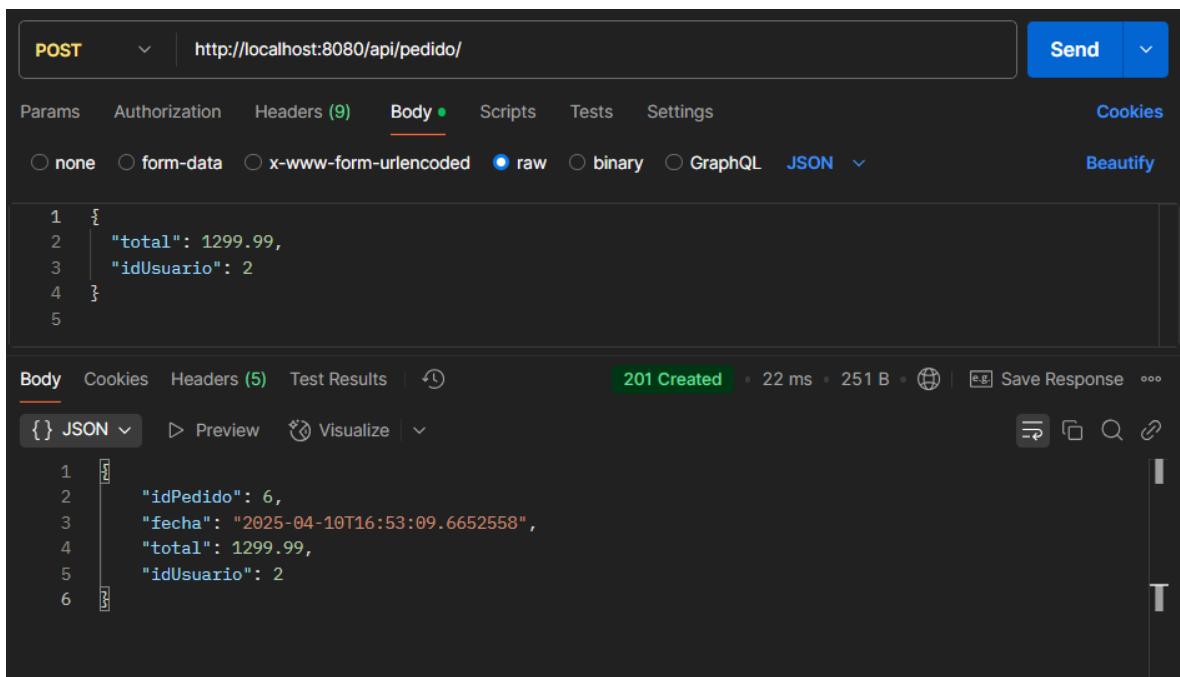
Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results 200 OK • 20 ms • 238 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "idPedido": 1,
3   "fecha": "2025-03-07T14:00:00",
4   "total": 1299.99,
5   "idUsuario": 1
6 }
```

POST <http://localhost:8080/api/pedido/>



POST <http://localhost:8080/api/pedido/> Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "total": 1299.99,
3   "idUsuario": 2
4 }
```

Body Cookies Headers (5) Test Results 201 Created • 22 ms • 251 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "idPedido": 6,
3   "fecha": "2025-04-10T16:53:09.6652558",
4   "total": 1299.99,
5   "idUsuario": 2
6 }
```

PUT <http://localhost:8080/api/pedido/6>

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

- Method:** PUT
- URL:** <http://localhost:8080/api/pedido/6>
- Body Type:** raw
- Request Body:**

```
1 {  
2   "total": 1399.99,  
3   "idUsuario": 2  
4 }  
5
```
- Response:** 200 OK, 28 ms, 245 B
- Response Body (JSON):**

```
1 {  
2   "idPedido": 6,  
3   "fecha": "2025-04-10T16:53:09.665255",  
4   "total": 1399.99,  
5   "idUsuario": 2  
6 }
```

DELETE <http://localhost:8080/api/pedido/6>

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

- Method:** DELETE
- URL:** <http://localhost:8080/api/pedido/6>
- Body Type:** Raw
- Response:** 204 No Content, 16 ms, 112 B

Below the response, there is a table for Query Params:

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

### 1. Configuraciones necesarias para la aplicación de consumo:

- La API debe estar en ejecución en la dirección `http://localhost:8080`.
- El puerto 8080 debe estar libre y disponible.

### 2. Cómo acceder a la sección del consumo del RESTful:

- Puede utilizarse una herramienta como Postman para realizar peticiones a los endpoints de la API.
- Alternativamente, se puede acceder a **Swagger UI**, una interfaz visual que permite interactuar fácilmente con la API desde el navegador.
- URL de acceso: <http://localhost:8080/swagger-ui.html>

### 3. Otras indicaciones importantes:

- Todos los datos deben enviarse en formato **JSON**.
- Los campos obligatorios están validados mediante anotaciones como `@NotBlank`, `@Email`, `@NotNull`.
- Las respuestas de error incluyen detalles específicos sobre los campos inválidos.
- Las fechas se manejan en formato ISO (`yyyy-MM-dd'T'HH:mm:ss`).
- Se puede usar Swagger UI para realizar pruebas sin necesidad de herramientas externas.