

# Desarrollo de la API tienda online: por Luis Baquiaux

API TIENDA ONLINE .....	2
Pasos de la creación y configuración del proyecto. ....	2
Implementación: .....	2
USO DEL MÉTODO GET .....	2
1. Lista de clientes: .....	2
2. Lista de de productos: .....	3
3. Producto por id: .....	4
4. Lista de carritos: .....	5
5. Lista productos de un carrito: .....	6
USO DEL MÉTODO POST .....	7
1. Insertar cliente .....	7
2. Insertar producto .....	8
3. Insertar carrito .....	9
4. Insertar producto a carrito .....	10
USO DEL MÉTODO PUT .....	11
1. Actualizar cliente .....	11
2. Actualizar producto .....	12
USO DEL MÉTODO DELETE .....	13
1. Eliminar producto por id: .....	13
2. Eliminar un producto de un carrito: .....	14
3. Vaciar un carrito por id: .....	15

# API TIENDA ONLINE

**Observación** se adjunta el código y los scripts necesarios para la creación de la base de datos y las tablas.

## ***Pasos de la creación y configuración del proyecto.***

1. Script de creación de la base de datos y las tablas.
2. Creación del proyecto con js: `npm init -y`
3. Instalación de dependencias: `npm i mysql2 express nodemon body-parser`
4. Configuración de la base de datos
  - a) Base de datos: `tienda_online`
  - b) Usuario: `root`
  - c) Contraseña: `luisbaquiaux`
5. Conexión a la base de datos
6. Configuración del puerto donde va estar “escuchado” la API a los visitantes: `PORT = 5000`
7. A través de express se usa los métodos `get`, `put`, `delete`, `post` donde estable las rutas que puede ser accedidas desde el exterior y realizar ciertas acciones, en este caso la conexión a la base de datos heladeria y la ejecución de script sql.

## ***Implementación:***

Para obtener un listado de entidades se usó el método `get`, para ingresar nuevos datos se usó `post`, para actualizar datos se utilizó el método `put` y para eliminar registros se usó el método `delete`.

## **USO DEL MÉTODO GET**

1. Lista de clientes:

api tienda online / **clientes** Save ▼ Sh

**GET** ▼  Send

Params Auth Headers (6) Body Scripts Settings Co

Query Params

	Key	Value	Description	...	Bulk
	Key	Value	Description		

Body ▼ ↺ 200 OK • 25 ms • 1.19 KB • 🌐 e.g.

Pretty Raw Preview Visualize JSON ▼ 🔗 📄

```
30  "correo": "ana.ramirez@gmail.com",
31  "telefono": "5553456",
32  "password": "hashed_password4",
33  "estado": 1
34  },
35  {
36    "nit": "55667788",
37    "nombre1": "Luis",
38    "nombre2": "Alberto",
39    "apellido1": "Morales",
40    "apellido2": "Castro",
41    "correo": "luis.morales@gmail.com",
42    "telefono": "5557890",
43    "password": "hashed_password5",
44    "estado": 1
45  },
46  {
47    "nit": "87654321",
48    "nombre1": "María",
49    "nombre2": "Elena",
50    "apellido1": "López",
```

2. Lista de de productos:

api tienda online / **productos** Save Share

**GET** http://localhost:5000/productos Send

Params Auth Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body 200 OK • 19 ms • 794 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔗 📋 🔍

```
8  {
9    "estado": 1,
10   },
11   {
12     "id": 2,
13     "nombre": "Frijoles",
14     "descripcion": "Alimento rico en proteínas",
15     "costo": 1.5,
16     "existencias": 80,
17     "estado": 1
18   },
19   {
20     "id": 3,
21     "nombre": "Azúcar",
22     "descripcion": "Endulzante de uso diario",
23     "costo": 0.9,
24     "existencias": 120,
25     "estado": 1
26   },
27   }
```

3. Producto por id:

GET ⌵ http://localhost:5000/productos/1 Send ⌵

Params Auth Headers (6) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	⋮ Bulk Edit
	Key	Value	Description	

Body ⌵ 🔄 200 OK • 11 ms • 347 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON ⌵ 🔄 🔗 📄 🔍

```
1  [  
2    {  
3      "id": 1,  
4      "nombre": "Arroz",  
5      "descripcion": "Grano básico para comidas",  
6      "costo": 1.2,  
7      "existencias": 100,  
8      "estado": 1  
9    }  
10 ]
```

4. Lista de carritos:

api tienda online / **carritos** Save Share

**GET** http://localhost:5000/carritos Send

Params Auth Headers (6) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body 200 OK • 7 ms • 1.13 KB • 🌐 📄 ...

Pretty Raw Preview Visualize **JSON** 🔍

```
1  [  
2    {  
3      "id": 1,  
4      "fecha": "2024-12-07T06:00:00.000Z",  
5      "total": 10.5,  
6      "nit_cliente": "12345678",  
7      "estado": 1  
8    },  
9    {  
10     "id": 2,  
11     "fecha": "2024-12-07T06:00:00.000Z",  
12     "total": 8.8,  
13     "nit_cliente": "12345678",  
14     "estado": 1  
15   },  
16   {  
17     "id": 3,  
18     "fecha": "2024-12-07T06:00:00.000Z",  
19     "total": 12.6,  
20     "nit_cliente": "87654321",  
21     "estado": 1
```

Console Postbot Runner Vault 11:46 09/12/2024 ENG 7

5. Lista productos de un carrito:

api tienda onli... / productos de un carrito y un cliente en específico

GET http://localhost:5000/carrito/1/12345678

Send

Params Auth Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body 200 OK 41 ms 550 B

Pretty Raw Preview Visualize JSON

```
[
  {
    "nombre": "Arroz blanco",
    "descripcion": "Grano básico para comidas",
    "cantidad": 2,
    "precio": 3,
    "subtotal": 6
  },
  {
    "nombre": "Frijoles",
    "descripcion": "Alimento rico en proteínas",
    "cantidad": 2,
    "precio": 4,
    "subtotal": 8
  },
  {
    "nombre": "Azúcar",
    "descripcion": "Endulzante de uso diario",
    "cantidad": 1,
    "precio": 1.5,
    "subtotal": 1.5
  }
]
```

## USO DEL MÉTODO POST

1. Insertar cliente

api tienda online / nuevo cliente

POST http://localhost:5000/insertar/cliente

Params Auth Headers (8) Body Scripts Settings Cookies Beautify

raw JSON

```
1 {
2   "nit": "12121212",
3   "nombre1": "Pedro",
4   "nombre2": "Neto",
5   "apellido1": "Sánchez",
6   "apellido2": "Hernández",
7   "correo": "pedro.sanchez@gmail.com",
8   "telefono": "5559012",
9   "password": "hashed_password3",
10  "estado": 1
11 }
```

Body 200 OK • 30 ms • 258 B • e.g. ...

Pretty Raw Preview Visualize HTML

```
1 Datos insertados correctamente
```

2. Insertar producto



api tienda online / nuevo producto

POST http://localhost:5000/insertar/producto

Params Auth Headers (8) Body Scripts Settings Cookies

raw JSON Beautify

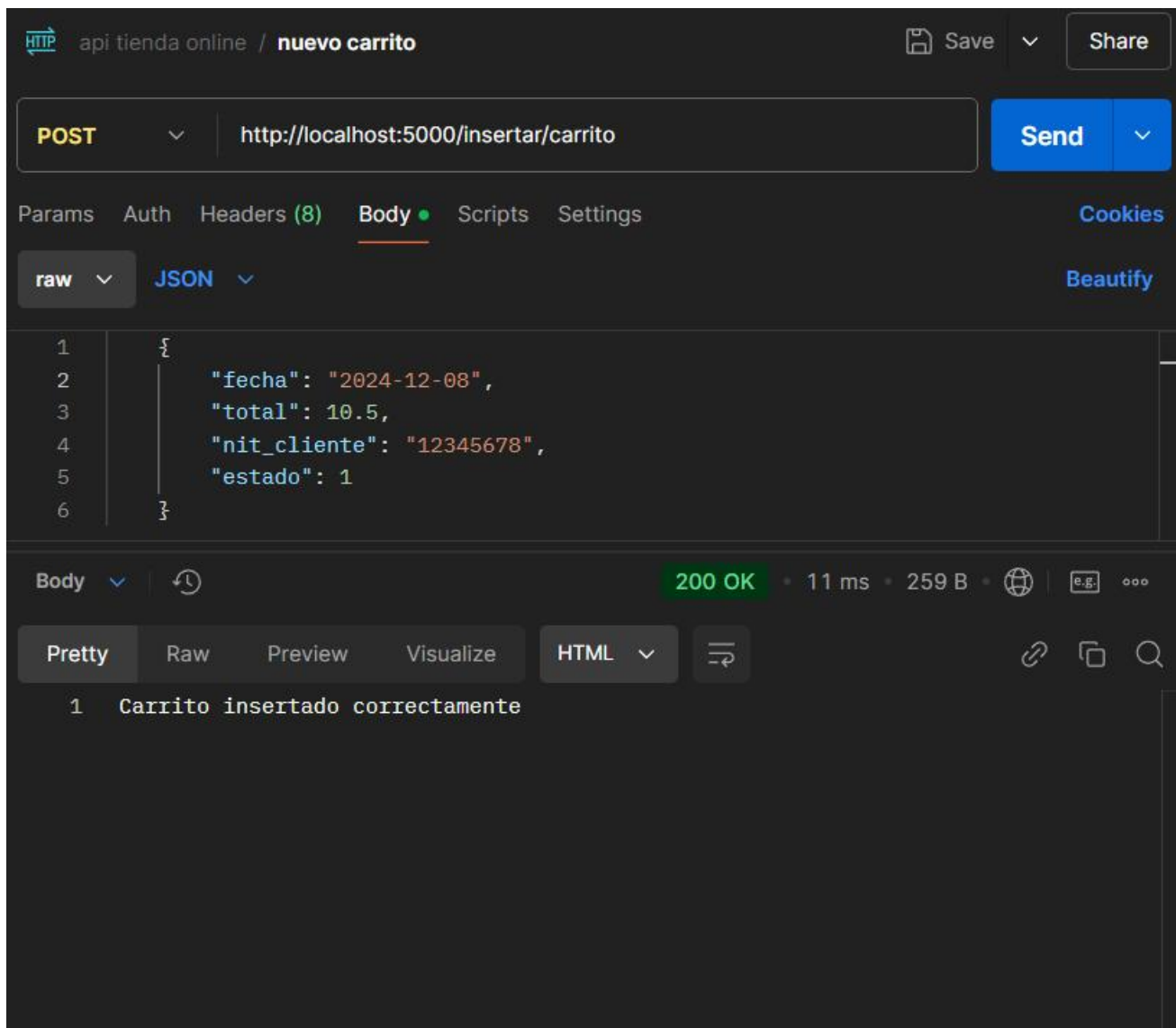
```
1 {
2   "nombre": "Nuevo producto",
3   "descripcion": "Este es un nuevo producto",
4   "costo": 10,
5   "existencias": 85,
6   "estado": 1
7 }
```

Body 200 OK • 30 ms • 260 B

Pretty Raw Preview Visualize HTML

```
1 Producto insertado correctamente
```

3. Insertar carrito



4. Insertar producto a carrito

api tienda online / agregar producto al carrito

POST http://localhost:5000/carrito/agregar

Send

Params Auth Headers (8) Body Scripts Settings Cookies Beautify

raw JSON

```
1 {
2   "id_carrito": 1,
3   "id_producto": 5,
4   "cantidad": 2,
5   "precio": 3,
6   "estado": 1
7 }
```

Body 200 OK • 71 ms • 265 B

Pretty Raw Preview Visualize HTML

```
1 Se ha agregado el producto al carrito
```

## USO DEL MÉTODO PUT

1. Actualizar cliente

api tienda online / nuevo cliente

POST http://localhost:5000/insertar/cliente

Send

Params Auth Headers (8) Body Scripts Settings Cookies

raw JSON Beautify

```
1 {
2   "nit": "12121213",
3   "nombre1": "Pedro",
4   "nombre2": "Neto",
5   "apellido1": "Sánchez",
6   "apellido2": "Hernández",
7   "correo": "pedro.sanchez@gmail.com",
8   "telefono": "5559012",
9   "password": "hashed_password3",
10  "estado": 1
11 }
```

Body 200 OK • 22 ms • 258 B

Pretty Raw Preview Visualize HTML

1 Datos insertados correctamente

2. Actualizar producto

api tienda online / **actualizar producto** Save Share

**PUT** ⌵  Send ⌵

Params Auth Headers (8) **Body** • Scripts Settings Cookies

**raw** ⌵ **JSON** ⌵ Beautify

```
1 {
2   "nombre": "Arroz blanco",
3   "descripcion": "Grano básico para comidas",
4   "costo": 1.2,
5   "existencias": 100,
6   "estado": 1
7 }
```

Body ⌵ ↺ 200 OK • 33 ms • 262 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize **HTML** ⌵ 🔗 📄 🔍

```
1 Producto actualizado correctamente
```

## USO DEL MÉTODO DELETE

1. Eliminar producto por id:

api tienda online / **eliminar producto** Save Share

**DELETE** ▼  Send ▼

Params Auth Headers (6) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body ▼ ↺ 200 OK • 13 ms • 260 B • 🌐 📄 ...

Pretty Raw Preview Visualize **HTML** ▼ ↺ 🔗 📄 🔍

```
1 Producto eliminado correctamente
```

2. Eliminar un producto de un carrito:

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

- URL Bar:** Displays the URL `http://localhost:5000/carrito/eliminar/23` with a "Save" button and a dropdown arrow.
- Method and URL:** A dropdown menu shows "DELETE" and the URL `http://localhost:5000/carrito/eliminar/23` is entered. A "Send" button is to the right.
- Params Tab:** The "Params" tab is selected, showing a "Query Params" section with a table:

	Key	Value	Description	...	Bulk t
	Key	Value	Description		

Below the table is a large empty text area.

- Response Section:** Shows the response status `200 OK` in green, with details `36 ms`, `272 B`, and a globe icon.
- Body Tab:** The "Body" tab is selected, showing the response body in "Pretty" format:

```
1  Producto eliminado del carrito correctamente
```

3. Vaciar un carrito por id:



api tienda online / vaciar carrito



Save



Share

DELETE



http://localhost:5000/carrito/vaciar/10

Send



Params

Auth

Headers (6)

Body

Scripts

Settings

Cookies

### Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body



200 OK

• 21 ms • 266 B

e.g.

...

Pretty

Raw

Preview

Visualize

HTML



```
1 Se ha vaciado el carrito correctamente
```