



SEMINARIO DE COMPLEMENTACIÓN PRÁCTICA III

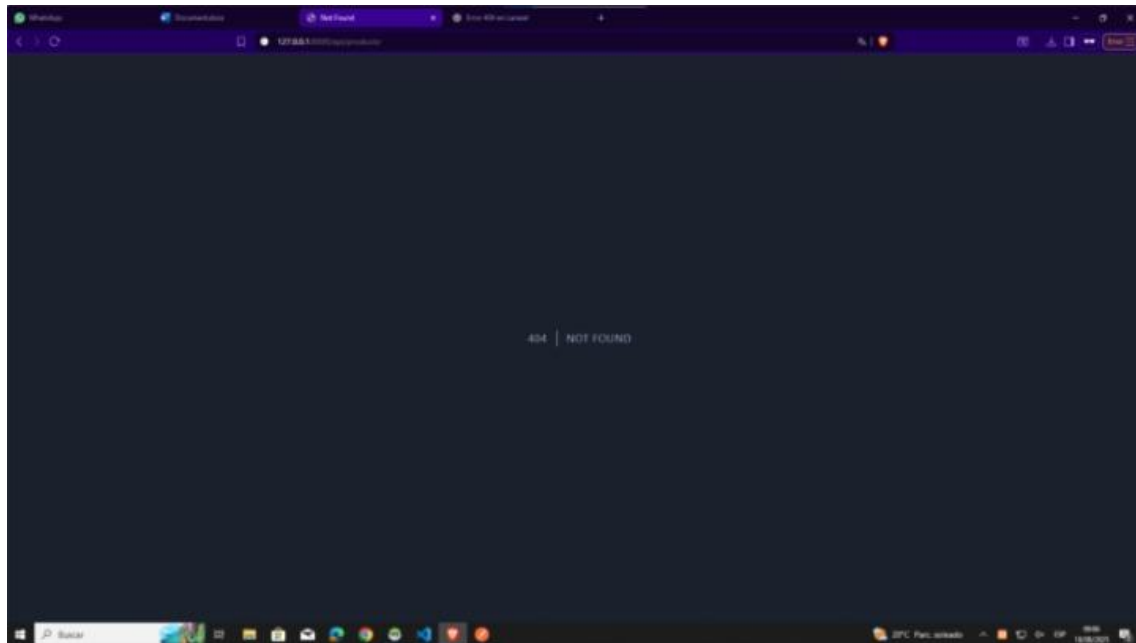


18 DE AGOSTO DE 2025

GIANFRANCO HUILLCA JAIMEZ

Mi casa

DOCUMENTAR LA SOLUCION AL ERROR 400



La solución fue la de instalar api desde el Git

```
alu_torre1@08CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan install:api
/composer.json has been updated
```

Y Con eso agregamos el

```
<?php

use App\Http\Controllers\Api\ProductController;

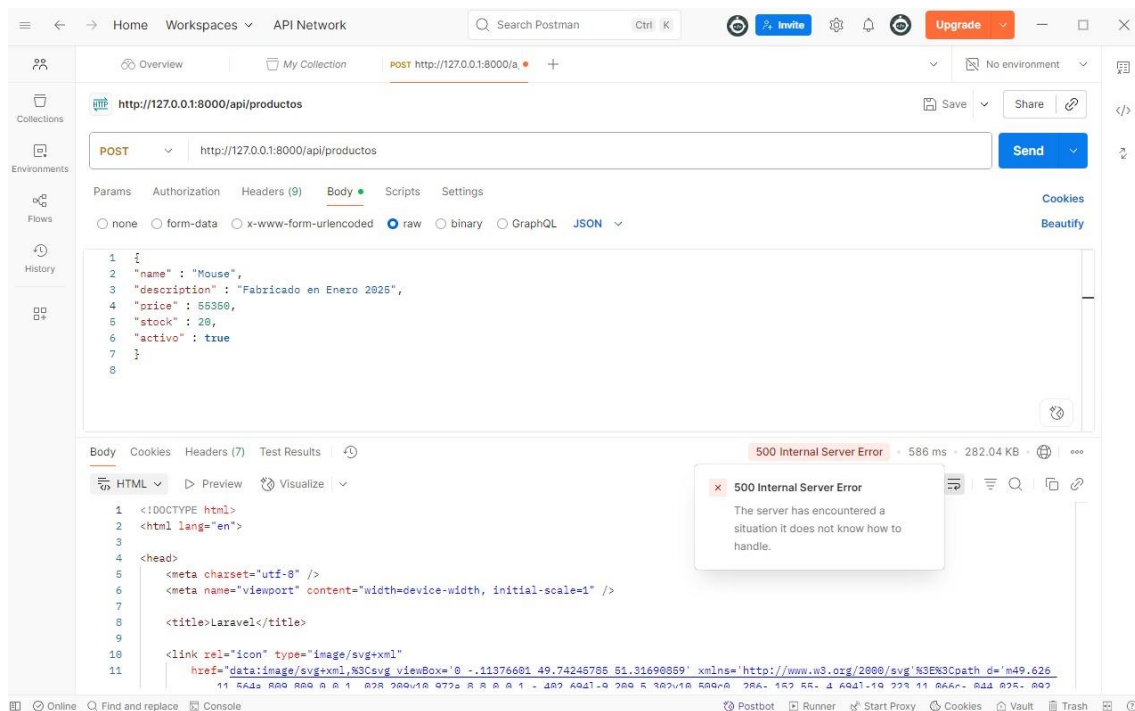
Route::apiResource('products', ProductController::class);
```

Y con eso sale asi

A screenshot of a web browser window showing the response of an API endpoint. The address bar shows the URL "127.0.0.1:8000/api/products". The page content is a JSON array of product data. The text is as follows:

```
{
  "data": [
    {
      "id": 1,
      "name": "mouse",
      "description": null,
      "sku": "pee",
      "stock": 30,
      "price": "55.60",
      "active": 1,
      "create_at": "2025-08-18T16:08:39.000000Z",
      "updated_at": "2025-08-18T16:08:39.000000Z"
    },
    {
      "id": 2,
      "name": "teclado",
      "description": null,
      "sku": "key123",
      "stock": 50,
      "price": "75.99",
      "active": 1,
      "create_at": "2025-08-18T16:18:26.000000Z",
      "updated_at": "2025-08-18T16:18:26.000000Z"
    },
    {
      "id": 3,
      "name": "monitor",
      "description": null,
      "sku": "mon456",
      "stock": 20,
      "price": "150.00",
      "active": 1,
      "create_at": "2025-08-18T16:18:48.000000Z",
      "updated_at": "2025-08-18T16:18:48.000000Z"
    },
    {
      "id": 4,
      "name": "auriculares",
      "description": null,
      "sku": "head789",
      "stock": 100,
      "price": "85.50",
      "active": 1,
      "create_at": "2025-08-18T16:18:58.000000Z",
      "updated_at": "2025-08-18T16:18:58.000000Z"
    },
    {
      "id": 5,
      "name": "webcam",
      "description": null,
      "sku": "cam321",
      "stock": 35,
      "price": "45.00",
      "active": 1,
      "create_at": "2025-08-18T16:19:46.000000Z",
      "updated_at": "2025-08-18T16:19:46.000000Z"
    }
  ]
}
```

DOCUMENTAR LA SOLUCION AL ERROR 500



Para la solución de el error 500 cambiamos el parámetro de `StoreProductoRequest` y cambiamos `El ProductoController.php`

```
public function store(StoreProductsRequest $request)
{
    $data = $request->validated();
    $product = Products::create($data);
    return response()->json($product, 201);
}
```

Ya con eso vemos en el HeidiSQL se guardan los datos correspondientes.

#	1 id	2 name	3 descripcion	4 sku	5 stock	6 price	7 active	8 deleted_at	9 created_at	10 updated_at
1	1	mouse	Fabricado en Enero 2025	pee	30	55.6	1	(NULL)	2025-08-18 16:08:39	2025-08-18 16:08:39
2	2	teclado	Teclado mecánico retroiluminado	key123	50	75.99	1	(NULL)	2025-08-18 16:18:26	2025-08-18 16:18:26
3	3	monitor	Monitor 24 pulgadas Full HD	mon456	20	150.0	1	(NULL)	2025-08-18 16:18:48	2025-08-18 16:18:48
4	4	auriculares	Auriculares inalámbricos con cancelación de ruido	head789	100	85.5	1	(NULL)	2025-08-18 16:18:58	2025-08-18 16:18:58
5	5	webcam	Cámara web HD para videoconferencias	cam321	35	45.0	1	(NULL)	2025-08-18 16:19:46	2025-08-18 16:19:46

DOCUMENTAR LAS PETICIONES GET,POST,PUT y DELETE

GET

GET

http://127.0.0.1:8000/api/products

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Cookies

Beautify

1

Ctrl+Alt+P for Postbot

Body

Cookies

Headers (7)

Test Results

200 OK

165 ms

1.12 KB

{}

JSON

Preview

Visualize

```
1 {
2   "data": [
3     {
4       "id": 1,
5       "name": "mouse",
6       "description": null,
7       "sku": "pee",
8       "stock": 30,
9       "price": "55.60",
10      "active": 1,
11      "create_at": "2025-08-18T16:08:39.000000Z",
12      "updated_at": "2025-08-18T16:08:39.000000Z"
13    },
14    {
15      "id": 2,
16      "name": "teclado",
17      "description": null,
18      "sku": "key123",
```

POST

POST

http://127.0.0.1:8000/api/products

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Cookies

Beautify

1

2

3

4

5

6

7

8

9

10

11

"id": 1,

"name": "mouse",

"description": null,

"sku": "pee",

"stock": 30,

"price": "55.60",

"active": 1,

"create_at": "2025-08-18T16:08:39.000000Z",

"updated_at": "2025-08-18T16:08:39.000000Z"

Body

Cookies

Headers (7)

Test Results

200 OK

165 ms

1.12 KB

{ } JSON

Preview

Visualize

INGRESA LA TABLA DE REGISTROS PRODUCTOS PEDIDOS Y CATEGORIAS Y CLIENTES

PRODUCTOS

products (Sr x 10q)										
#	1 id	2 name	3 descripcion	4 sku	5 stock	6 price	7 active	8 deleted_at	9 created_at	10 updated_at
1	1	mouse	Fabricado en Enero 2025	pee	30	55.6	1	(NULL)	2025-08-18 16:08:39	2025-08-18 16:08:39
2	2	teclado	Teclado mecánico retroiluminado	key123	50	75.99	1	(NULL)	2025-08-18 16:18:26	2025-08-18 16:18:26
3	3	monitor	Monitor 24 pulgadas Full HD	mon456	20	150.0	1	(NULL)	2025-08-18 16:18:48	2025-08-18 16:18:48
4	4	auriculares	Auriculares inalámbricos con cancelación de ruido	head789	100	85.5	1	(NULL)	2025-08-18 16:18:58	2025-08-18 16:18:58
5	5	webcam	Cámara web HD para videollamadas	cam321	35	45.0	1	(NULL)	2025-08-18 16:19:46	2025-08-18 16:19:46

PEDIDOS

#	1 id	2 order_date	3 price	4 status	5 notes	6 deleted_at	7 created_at	8 updated_at
1	1	2025-08-18 15:30:00	120.5	pending	Entrega urgente, revisar embalaje	(NULL)	2025-08-18 16:51:36	2025-08-18 16:51:36
2	2	2025-08-19 09:30:00	320.0	processing	Urgente, revisar stock antes de enviar	(NULL)	2025-08-18 16:51:58	2025-08-18 16:51:58
3	3	2025-08-20 16:15:00	45.5	completed	Entrega finalizada y confirmada	(NULL)	2025-08-18 16:52:06	2025-08-18 16:52:06
4	4	2025-08-22 11:45:00	80.0	cancelled	Cancelada por solicitud del cliente	(NULL)	2025-08-18 16:52:17	2025-08-18 16:52:17
5	5	2025-08-25 08:00:00	99.99	pending	(NULL)	(NULL)	2025-08-18 16:52:23	2025-08-18 16:52:23