

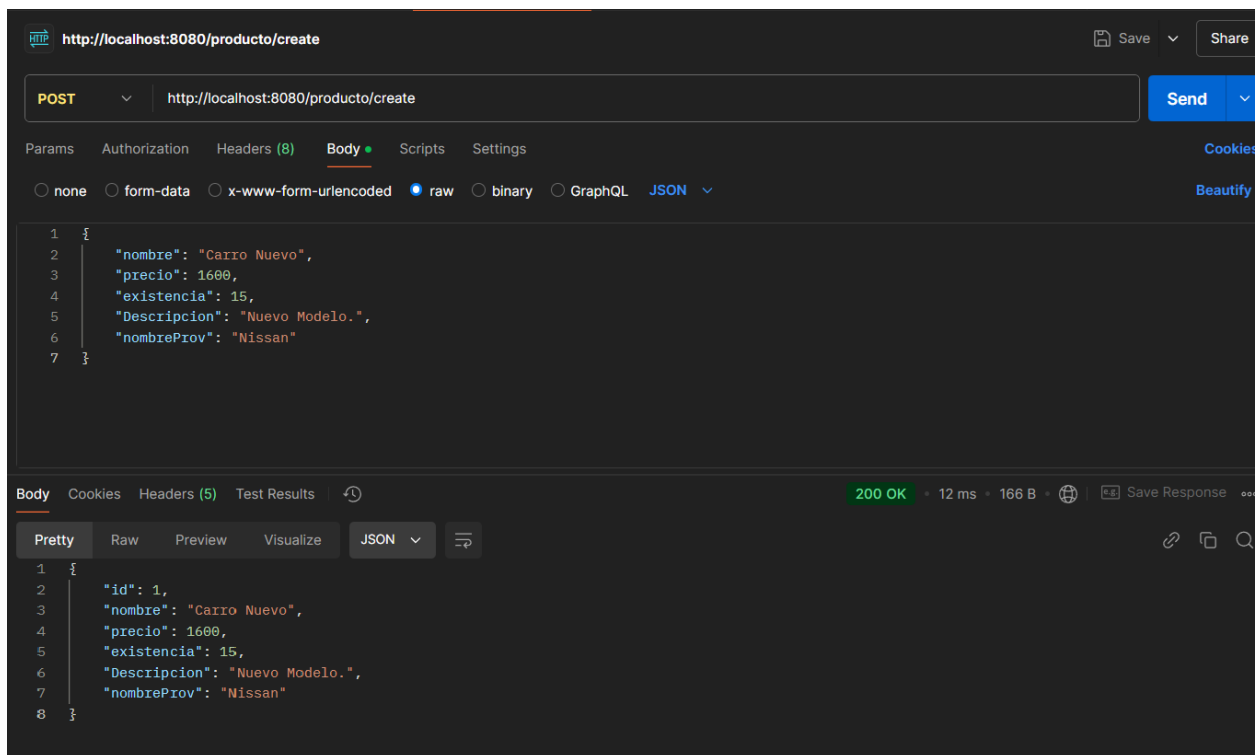
Victor Adrian Quintero Mata

2-4

18/11/2024

Examen 2—

- Verificar los endpoint de la aplicación mediante un gestor de peticiones HTTP en Postman.
- Comprobar la creación del registro de producto utilizando el método GET creado en la API.
- Modificar un usuario mediante el método HTTP PUT seleccionando un producto a modificar mediante el ID en la ruta de la api y pasándole un producto en el body.
- Visualiza la modificación buscando todos los productos de la base de datos de ventas mediante el método HTTP GET.
- Buscar el producto por el ID para mostrar el registro.
- Subir el proyecto (carpeta) al repositorio de Github.
- Subir a Github imagenes de evidencia del funcionamiento en postman.



HTTP client interface showing a POST request to `http://localhost:8080/producto/create`. The request body is a JSON object:

```
1 {
2   "nombre": "Carro SemiNuevo",
3   "precio": 1100,
4   "existencia": 10,
5   "Descripcion": "SemiNuevo.",
6   "nombreProv": "Nissan"
7 }
```

The response is a 200 OK status, 12 ms, 166 B. The response body is a JSON object:

```
1 {
2   "id": 2,
3   "nombre": "Carro Seminuevo",
4   "precio": 1100,
5   "existencia": 10,
6   "Descripcion": "Seminuevo.",
7   "nombreProv": "Nissan"
8 }
```

GET ALL

HTTP client interface showing a GET request to `http://localhost:8080/producto/all`. The response is a 200 OK status, 12 ms, 166 B. The response body is a JSON array of two objects:

```
1 [
2   {
3     "id": 1,
4     "nombre": "Carro Nuevo",
5     "precio": 1600,
6     "existencia": 15,
7     "Descripcion": "Nuevo Modelo.",
8     "nombreProv": "Nissan"
9   },
10  {
11    "id": 2,
12    "nombre": "Carro Seminuevo",
13    "precio": 1100,
14    "existencia": 10,
15    "Descripcion": "Seminuevo.",
16    "nombreProv": "Nissan"
17  }
18 ]
```

GET BY ID

http://localhost:8080/producto/1

GET http://localhost:8080/producto/1

Params Authorization Headers (8) Body Scripts Settings

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

```
1 {
2   "nombre": "Carro Seminuevo",
3   "precio": 1100,
4   "existencia": 10,
5   "Descripcion": "Seminuevo.",
6   "nombreProv": "Nissan"
7 }
```

Body Cookies Headers (5) Test Results

200 OK 9 ms 166 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "nombre": "Carro Nuevo",
4   "precio": 1600,
5   "existencia": 15,
6   "Descripcion": "Nuevo Modelo.",
7   "nombreProv": "Nissan"
8 }
```

UPDATE BY ID

http://localhost:8080/update/2

PUT http://localhost:8080/update/2

Params Authorization Headers (8) Body Scripts Settings

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

```
1 {
2   "nombre": "Carro Viejo",
3   "precio": 900,
4   "existencia": 2,
5   "Descripcion": "Modelo Viejo.",
6   "nombreProv": "Nissan"
7 }
```

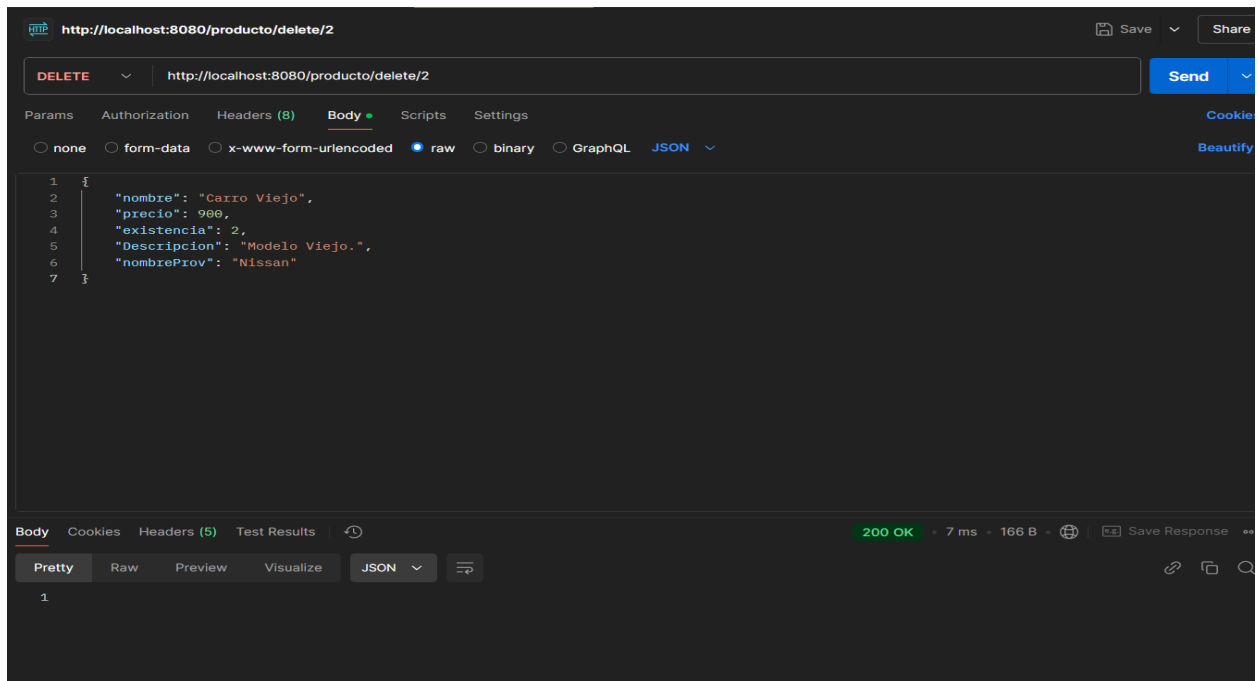
Body Cookies Headers (5) Test Results

200 OK 9 ms 166 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "nombre": "Carro Viejo",
4   "precio": 900,
5   "existencia": 2,
6   "Descripcion": "Modelo Viejo.",
7   "nombreProv": "Nissan"
8 }
```

DELETE BY ID



BASE DE DATOS EN TABLA PRODUCTOS

