

Taller 14 de Febrero 2025

Andrés Felipe Muñoz Aguilar
2210087
Grupo E1

Operación GET en ThunderClient

```
@RestController
@RequestMapping("/productos")
public class ProductoController {

    @Autowired    Remove this field injection a
    private ProductoService productoService;

    @GetMapping("/list")
    public List<Producto> cargarUsuarios() {
        return productoService.getProductos();
    }
}
```

The screenshot shows the ThunderClient interface. The top bar indicates a GET request to `localhost:8094/productos/list` with a status of **200 OK**, size of **462 Bytes**, and time of **14 ms**. The **Body** tab is selected, showing the JSON response. The response is a list of two product objects, each with a provider, name, price, and address.

Query: GET `localhost:8094/productos/list` **Send**

Status: 200 OK **Size:** 462 Bytes **Time:** 14 ms

Response:

```
[
  {
    "id": 1,
    "proveedor": {
      "id": 2,
      "ciudad": "Bogotá",
      "direccion": "Carrera 123 # 12 - 80",
      "nombre": "Colgate",
      "telefono": "312456123",
      "nit": "890123122"
    },
    "ivaCompra": 0.1,
    "nombre": "Miguel Bose",
    "precioCompra": 650000.0,
    "precioVenta": 655000.0
  },
  {
    "id": 2,
    "proveedor": {
      "id": 2,
      "ciudad": "Bogotá",
      "direccion": "Carrera 123 # 12 - 80",
      "nombre": "Colgate",
      "telefono": "312456123",
      "nit": "890123122"
    },
    "ivaCompra": 0.1,
    "nombre": "Miguel Bose",
    "precioCompra": 650000.0,
    "precioVenta": 655000.0
  }
]
```

Operación GET basado en ID en ThunderClient

```
public class ProductoController {  
  
    @GetMapping("/list/{id}")  
    public Producto buscarPorId(@PathVariable Long id) {  
        return productoService.buscarProducto(id);  
    }  
}
```

The screenshot shows the ThunderClient interface. The top bar indicates a GET request to `localhost:8094/productos/list/1` with a status of 200 OK, size of 230 Bytes, and time of 15 ms. The 'Body' tab is selected, showing the JSON response. The response is a JSON object with the following structure:

```
{  
  "id": 1,  
  "proveedor": {  
    "id": 2,  
    "ciudad": "Bogotá",  
    "direccion": "Carrera 123 # 12 - 80",  
    "nombre": "Colgate",  
    "telefono": "312456123",  
    "nit": "890123122"  
  },  
  "ivaCompra": 0.1,  
  "nombre": "Miguel Bose",  
  "precioCompra": 650000.0,  
  "precioVenta": 655000.0  
}
```

Operación POST en ThunderClient

```
public class ProductoController {  
  
    @PostMapping("/")  
    public ResponseEntity<Producto> agregar(@RequestBody Producto producto) {  
        Producto obj = productoService.nuevoProducto(producto);  
        return new ResponseEntity<>(obj, HttpStatus.OK);  
    }  
}
```

POST localhost:8094/productos/ Send

Status: 200 OK Size: 235 Bytes Time: 138 ms

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL

JSON Content Format

```
1 {
2   "proveedor": {
3     "id": 2,
4     "ciudad": "Bogotá",
5     "direccion": "Carrera 123 # 12 - 80",
6     "nombre": "Colgate",
7     "telefono": "312456123",
8     "nit": "890123122"
9   },
10  "ivaCompra": 0.45,
11  "nombre": "Miguel Jasón",
12  "precioCompra": 1230322,
13  "precioVenta": 4555382
14 }
```

Response Headers⁴ Cookies Results Docs

```
1 {
2   "id": 3,
3   "proveedor": {
4     "id": 2,
5     "ciudad": "Bogotá",
6     "direccion": "Carrera 123 # 12 - 80",
7     "nombre": "Colgate",
8     "telefono": "312456123",
9     "nit": "890123122"
10  },
11  "ivaCompra": 0.45,
12  "nombre": "Miguel Jasón",
13  "precioCompra": 1230322.0,
14  "precioVenta": 4555382.0
15 }
```

Operación PUT en ThunderClient

```
public class ProductoController {

    @PutMapping("/")
    public ResponseEntity<Producto> editar(@RequestBody Producto producto) {
        Producto obj = productoService.buscarProducto(producto.getId());
        if (obj != null) {

            obj.setId(producto.getId());
            obj.setProveedor(producto.getProveedor());
            obj.setIvaCompra(producto.getIvaCompra());
            obj.setNombre(producto.getNombre());
            obj.setPrecioCompra(producto.getPrecioCompra());
            obj.setPrecioVenta(producto.getPrecioVenta());

            productoService.nuevoProducto(obj);

            return new ResponseEntity<>(obj, HttpStatus.OK);
        } else {
            return new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }
}
```

PUT localhost:8094/productos/ Send

Status: 200 OK Size: 252 Bytes Time: 44 ms

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL

JSON Content Format

```
1 {
2   "id": 3,
3   "proveedor": {
4     "id": 2,
5     "ciudad": "Bogotá",
6     "direccion": "Carrera 123 # 12 - 80",
7     "nombre": "Colgate",
8     "telefono": "312456123",
9     "nit": "890123122"
10  },
11  "ivaCompra": 0.45,
12  "nombre": "Miguel Jasón Rentería Ocoró",
13  "precioCompra": 1230322,
14  "precioVenta": 4555382.0
15 }
```

Response Headers⁴ Cookies Results Docs

```
1 {
2   "id": 3,
3   "proveedor": {
4     "id": 2,
5     "ciudad": "Bogotá",
6     "direccion": "Carrera 123 # 12 - 80",
7     "nombre": "Colgate",
8     "telefono": "312456123",
9     "nit": "890123122"
10  },
11  "ivaCompra": 0.45,
12  "nombre": "Miguel Jasón Rentería Ocoró",
13  "precioCompra": 1230322.0,
14  "precioVenta": 4555382.0
15 }
```

Operación DELETE en ThunderClient

```
public class ProductoController {

    @DeleteMapping("/{id}")
    public ResponseEntity<Producto> eliminar(@PathVariable Long id) {
        Producto obj = productoService.buscarProducto(id);
        if (obj != null) {
            productoService.borrarProducto(id);
            return new ResponseEntity<>(obj, HttpStatus.OK);
        } else {
            return new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }
}
```

DELETE localhost:8094/productos/4 Send

Status: 200 OK Size: 233 Bytes Time: 50 ms

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL

JSON Content Format

```
1 {
2   "id": 4,
3   "proveedor": {
4     "id": 2,
5     "ciudad": "Bogotá",
6     "direccion": "Carrera 123 # 12 - 80",
7     "nombre": "Colgate",
8     "telefono": "312456123",
9     "nit": "890123122"
10  },
11  "ivaCompra": 0.85,
12  "nombre": "Jaimes Planta",
13  "precioCompra": 757789.0,
14  "precioVenta": 332376.0
15 }
```

Documentación con SWAGGER-UI

Swagger
supported by SMARTBEAR

/v3/api-docs Explore

OpenAPI definition ^{v0} OAS 3.1
[/v3/api-docs](#)

Servers

http://localhost:8094 - Generated server url

producto-controller

- PUT /productos/
- POST /productos/
- GET /productos/list
- GET /productos/list/{id}
- DELETE /productos/{id}

Ejemplo de Petición GET con springdoc-openapi-starter-webmvc-ui

Responses

Curl

```
curl -X 'GET' \  
  'http://localhost:8094/productos/list' \  
  -H 'accept: */*'
```

Request URL

```
http://localhost:8094/productos/list
```

Server response

Code	Details
200	<div>Response body<pre>[{ "id": 1, "proveedor": { "id": 2, "ciudad": "Bogotá", "direccion": "Carrera 123 # 12 - 80", "nombre": "Colgate", "telefono": "312456123", "nit": "890123122" }, "ivaCompra": 0.1, "nombre": "Miguel Bose", "precioCompra": 650000, "precioVenta": 655000 },]</pre></div>