

Listado de Peticiones REST - Proyecto Reto 1 - Clean Architecture

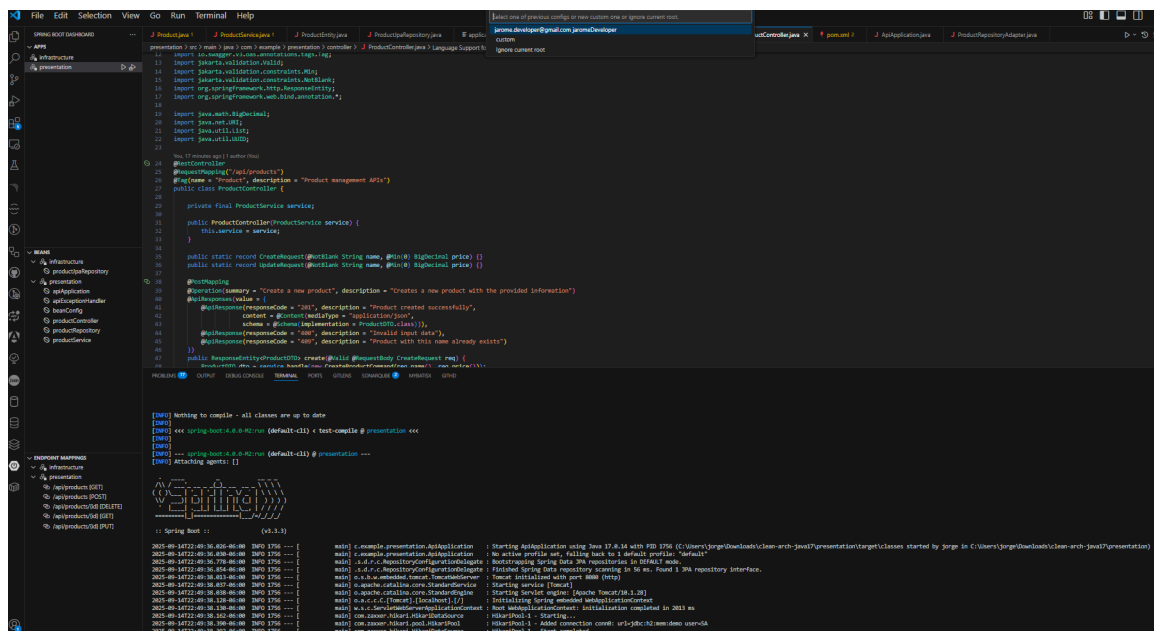
Este documento describe los endpoints disponibles en la API CRUD de Product, desarrollada con Java 17 + Spring Boot siguiendo principios de Clean Architecture.

Profesor: Jorge Valente Fragoso Mora

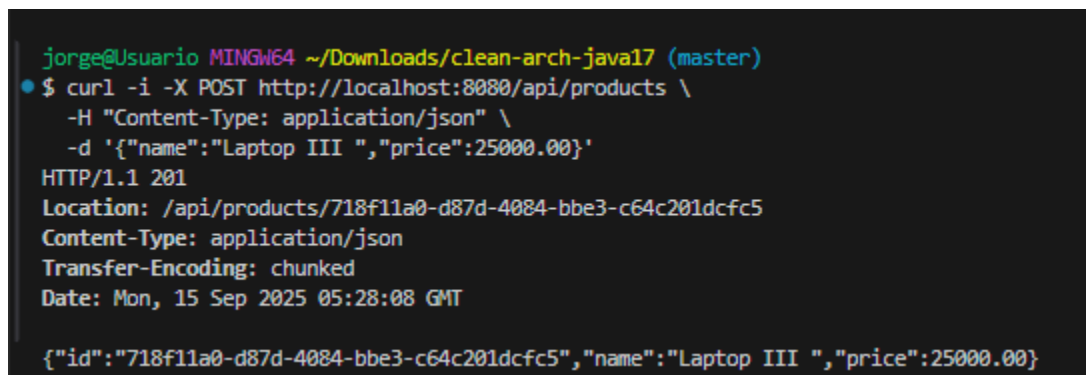
Alumno: Jorge Alberto Rodriguez Mendez

Correo: jarome.developer@gmail.com

El proyecto está ejecutándose en el ambiente local.



Crear un producto:



Listar todos los productos:

```
jorge@Usuario MINGW64 ~/Downloads/clean-arch-java17 (master)
$ curl -i http://localhost:8080/api/products
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Mon, 15 Sep 2025 05:28:52 GMT

[{"id":"5072283b-7c3b-43fc-8629-6450e199655d","name":"Laptop","price":25000.00},{id":"445db509-dbbe-4e5b-94c7-7f1bad30379b","name":"Laptop HPS","price":25000.00}]
```

Obtener un producto por ID:

```
jorge@Usuario MINGW64 ~/Downloads/clean-arch-java17 (master)
$ curl -i http://localhost:8080/api/products/b981106e-edb2-439d-a756-874f08d662f3
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Mon, 15 Sep 2025 05:52:30 GMT

{"id":"b981106e-edb2-439d-a756-874f08d662f3","name":"Laptop Gamer","price":28000.00}
```

Actualizar un producto

```
jorge@Usuario MINGW64 ~/Downloads/clean-arch-java17 (master)
$ curl -i -X POST http://localhost:8080/api/products -H "Content-Type: application/json" -d '{"name":"Laptop IV ","price":25000.00}'
HTTP/1.1 201
Location: /api/products/b981106e-edb2-439d-a756-874f08d662f3
Content-Type: application/json
Transfer-Encoding: chunked
Date: Mon, 15 Sep 2025 05:51:19 GMT

{"id":"b981106e-edb2-439d-a756-874f08d662f3","name":"Laptop IV ","price":25000.00}
jorge@Usuario MINGW64 ~/Downloads/clean-arch-java17 (master)
$ curl -i -X PUT http://localhost:8080/api/products/b981106e-edb2-439d-a756-874f08d662f3 -H "Content-Type: application/json" -d '{"name":"Laptop Gamer","price":28000.00}'
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Mon, 15 Sep 2025 05:51:39 GMT

{"id":"b981106e-edb2-439d-a756-874f08d662f3","name":"Laptop Gamer","price":28000.00}
```

Eliminar un producto

```
jorge@Usuario MINGW64 ~/Downloads/clean-arch-java17 (master)
$ curl -i -X DELETE http://localhost:8080/api/products/b981106e-edb2-439d-a756-874f08d662f3
HTTP/1.1 204
Date: Mon, 15 Sep 2025 05:53:58 GMT
```

Resumen de Endpoints

Método	Endpoint	Descripción
--------	----------	-------------

POST	/api/products	Crear un producto
GET	/api/products	Listar todos los productos
GET	/api/products/{id}	Obtener producto por ID
PUT	/api/products/{id}	Actualizar producto
DELETE	/api/products/{id}	Eliminar producto

1. Crear un producto

Método: POST

Endpoint: /api/products

Request Body (JSON):

```
{  
  "name": "Laptop",  
  "price": 25000.00  
}
```

Response Ejemplo:

```
{  
  "id": "d6e8f9a3-44f5-4a5b-9e8b-6cfc4dd0f19e",  
  "name": "Laptop",  
  "price": 25000.00  
}
```

2. Listar todos los productos

Método: GET

Endpoint: /api/products

Response Ejemplo:

```
[  
  {  
    "id": "uuid1",  
    "name": "Laptop",  
    "price": 25000.00  
  },  
  {  
    "id": "uuid2",  
    "name": "Mouse",  
    "price": 500.00  
  }  
]
```

```
}  
]
```

3. Obtener producto por ID

Método: GET

Endpoint: /api/products/{id}

Response Ejemplo:

```
{  
  "id": "uuid1",  
  "name": "Laptop",  
  "price": 25000.00  
}
```

Notas: Si el producto no existe, retorna 404 con body: {"error": "Product not found"}

4. Actualizar producto

Método: PUT

Endpoint: /api/products/{id}

Request Body (JSON):

```
{  
  "name": "Laptop Gamer",  
  "price": 28000.00  
}
```

Response Ejemplo:

```
{  
  "id": "uuid1",  
  "name": "Laptop Gamer",  
  "price": 28000.00  
}
```

5. Eliminar producto

Método: DELETE

Endpoint: /api/products/{id}

Response Ejemplo:

HTTP 204 No Content (sin body)
