

ANEXO SPRINGBOOT

Alumna: Pérez García, Carolina

Curso: 3K9

A continuación se presentan las capturas de pantalla del Proyecto de SpringBoot:

■ Documentación completa de endpoints

The screenshot shows a REST API documentation interface. At the top, there is a dropdown menu labeled "Servers" with the option "http://localhost:8080 - Generated server url". Below this, a section titled "Gestión de Productos" is shown, described as "API REST para gestionar productos". The section lists the following endpoints:

- GET /api/productos/{id}
- PUT /api/productos/{id}
- DELETE /api/productos/{id} (highlighted in red)
- GET /api/productos
- POST /api/productos
- PATCH /api/productos/{id}/stock
- GET /api/productos/categoría

The screenshot shows a REST API documentation interface. A section titled "Schemas" is displayed, containing the following schema definitions:

- ProductoDTO >
- ProductoResponseDTO >
- ActualizarStockDTO >

- Prueba exitosa de POST (creando producto)

http://localhost:8080/api/productos

Server response

Code	Details
201 <i>Undocumented</i>	Response body <pre>{ "id": 1, "nombre": "Laptop Gamer", "descripcion": "Laptop con RTX 4070", "precio": 1500.99, "stock": 10, "categoria": "Electrónica" }</pre> Copy Download
	Response headers <pre>connection: keep-alive content-type: application/json date: Mon,10 Nov 2025 02:32:16 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

Responses

Code	Description	Links
200	OK	No links

- Prueba de GET (listando productos)

Code	Details
200	<p>Response body</p> <pre>[{ "id": 1, "nombre": "Laptop Gamer", "descripcion": "Laptop con RTX 4070", "precio": 1500.99, "stock": 10, "categoria": "Electrónica" }, { "id": 2, "nombre": "Aspiradora Liliana", "descripcion": "Modelo A3-09", "precio": 999.2, "stock": 16, "categoria": "Hogar" }, { "id": 3, "nombre": "Pelota Dynasty Nro 5", "descripcion": "Color Amarillo", "precio": 678.9, "stock": 16, "categoria": "Deportes" }, { "id": 4, "nombre": "Vestido Caro Cuore", "descripcion": "Talla S", "precio": 1250.0, "stock": 12, "categoria": "Moda" }]</pre>

- Error 404 cuando producto no existe

http://localhost:8080/api/productos/7/stock	
Server response	
Code	Details
404	Error: response status is 404
	<i>Undocumented</i>
	Response body
	<pre>{ "timestamp": "2025-11-09T23:55:01.1699224", "estado": 404, "mensaje": "Producto no encontrado con id: 7", "ruta": "/api/productos/7/stock"}]</pre>

- Error 400 de validación

400 Error: response status is 400
Undocumented

Response body

```
{  
  "precio": "El precio debe ser como mínimo 0.01"  
}
```

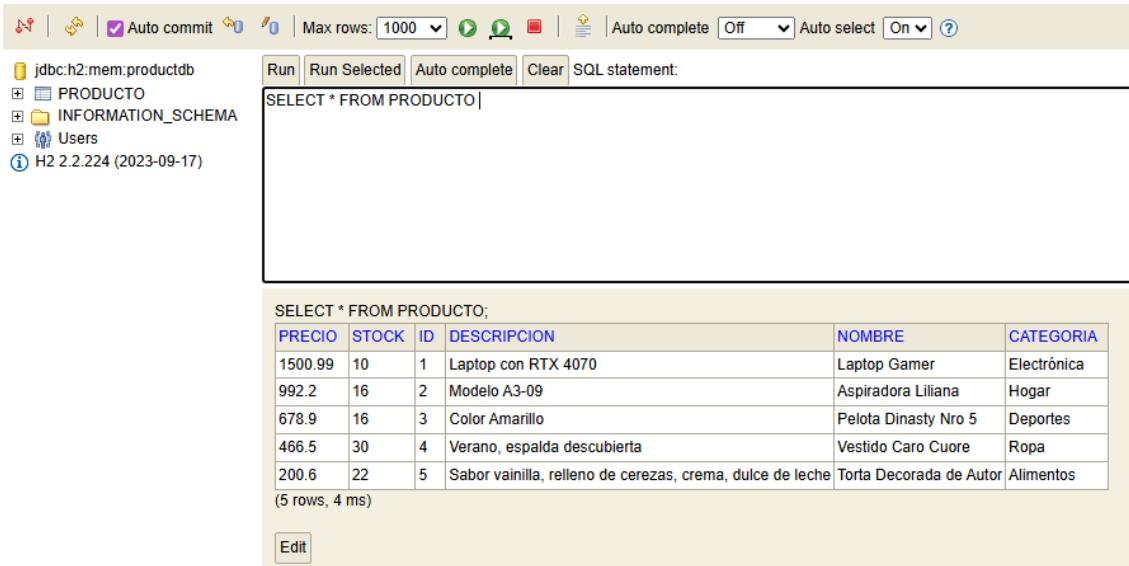
Download

Response headers

```
connection: close  
content-type: application/json  
date: Mon, 10 Nov 2025 02:48:37 GMT  
transfer-encoding: chunked
```

Responses

■ Consola H2 con datos persistidos



The screenshot shows the H2 Database Console interface. On the left, there's a sidebar with database connections and schema information:

- jdbc:h2:mem:productdb
- PRODUCTO
- INFORMATION_SCHEMA
- Users
- H2 2.2.224 (2023-09-17)

The main area has a toolbar at the top with various icons and dropdowns for settings like 'Auto commit', 'Max rows', 'Auto complete', and 'Auto select'. Below the toolbar, a SQL statement is entered:

```
SELECT * FROM PRODUCTO
```

The results of the query are displayed in a table:

PRECIO	STOCK	ID	DESCRIPCION	NOMBRE	CATEGORIA
1500.99	10	1	Laptop con RTX 4070	Laptop Gamer	Electrónica
992.2	16	2	Modelo A3-09	Aspiradora Liliana	Hogar
678.9	16	3	Color Amarillo	Pelota Dinasty Nro 5	Deportes
466.5	30	4	Verano, espalda descubierta	Vestido Caro Cuore	Ropa
200.6	22	5	Sabor vainilla, relleno de cerezas, crema, dulce de leche	Torta Decorada de Autor	Alimentos

(5 rows, 4 ms)

An 'Edit' button is located at the bottom of the results table.