



Creación de Una API (15%)

Autor: Jimmis J. Simanca

Tecnología en Desarrollo de Software, Corporación Universitaria Uniremington

Asignatura: Lenguaje de Programación 3

Director: Milton Javier Mateus Hernández

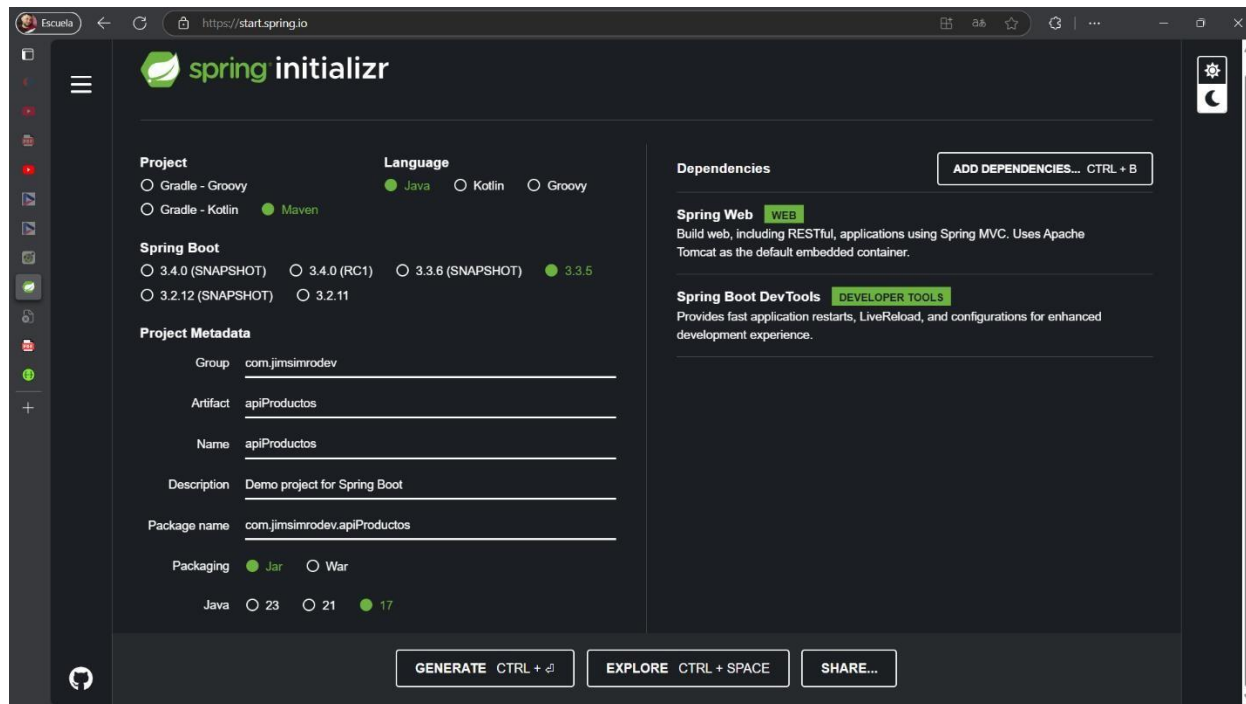
31 de octubre de 2024



UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Tabla de contenido	Creación del proyecto ¡Error! Marcador no definido.
Estructura del proyecto	3
Pruebas de los endpoints.....	4
Documentacion en swagger	7

Creación del proyecto



The screenshot shows the Spring Initializr web application interface. The browser address bar displays "https://start.spring.io". The page title is "spring inicializr". The interface is divided into several sections:

- Project:** Includes radio buttons for "Gradle - Groovy", "Gradle - Kotlin", and "Maven" (selected).
- Language:** Includes radio buttons for "Java" (selected), "Kotlin", and "Groovy".
- Spring Boot:** Includes radio buttons for "3.4.0 (SNAPSHOT)", "3.4.0 (RC1)", "3.3.6 (SNAPSHOT)", and "3.3.5" (selected).
- Project Metadata:** Includes input fields for "Group" (com.jimsimrodev), "Artifact" (apiProductos), "Name" (apiProductos), "Description" (Demo project for Spring Boot), and "Package name" (com.jimsimrodev.apiProductos).
- Packaging:** Includes radio buttons for "Jar" (selected) and "War".
- Java:** Includes radio buttons for "23", "21", and "17" (selected).
- Dependencies:** Includes a button "ADD DEPENDENCIES... CTRL + B" and two sections: "Spring Web" (WEB) and "Spring Boot DevTools" (DEVELOPER TOOLS).

At the bottom, there are three buttons: "GENERATE CTRL + G", "EXPLORE CTRL + SPACE", and "SHARE...".

Estructura del proyecto

```

34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
59
1
2
3
4

@Autowired
private ProductoServiceImpl productoServiceImpl;

/**
 * Endpoint Retorna la lista de los productos que hay en la lista y los muestra
 * en json
 */
@GetMapping
public List<Producto> getProdutos() {
    return productoServiceImpl.listarProductos();
}

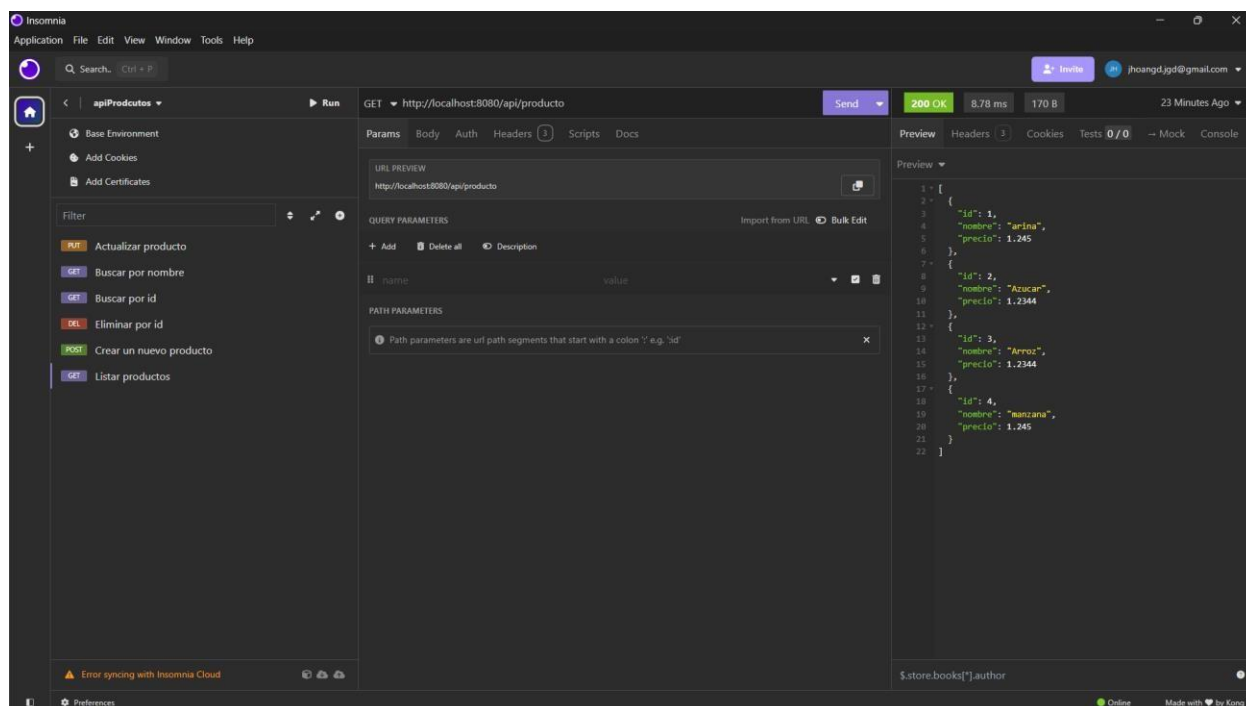
/**
 * Endpoint Realiza la busqueda por id y si no existe retorna un mensaje
 */
@GetMapping("/{id}")
public Producto buscarPorId(@PathVariable Long id) {
    var producto = productoServiceImpl.buscarPorId(id);
    return producto;
}

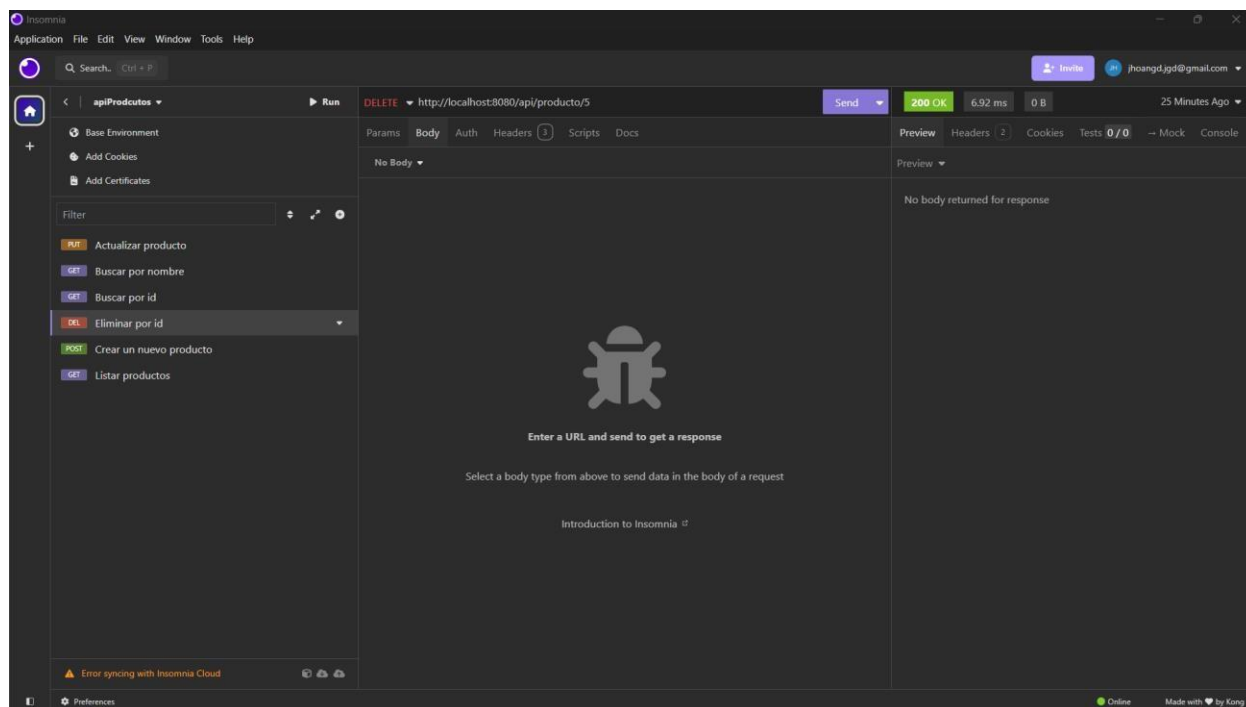
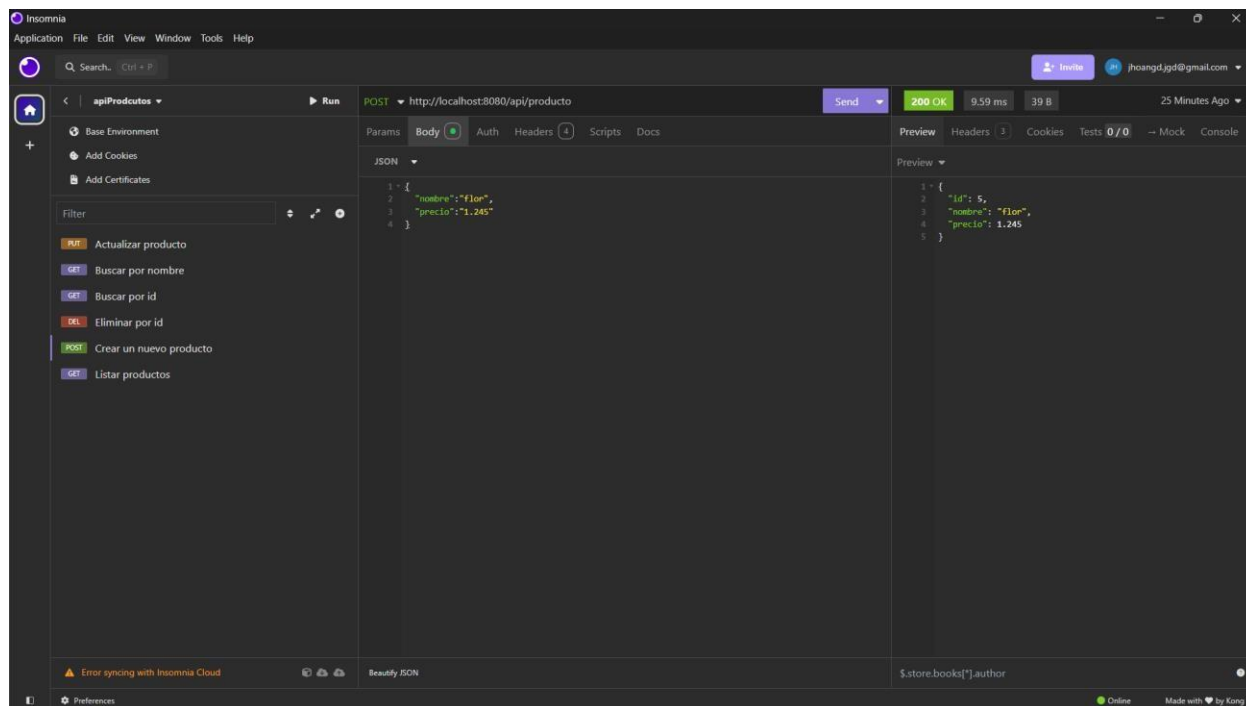
/**
 * Endpoint Realiza la busqueda por nombre y si no existe retorna un mensaje
 */
@GetMapping("/buscar")
public List<Producto> buscarPorNombre(@RequestParam String nombre) {
    return productoServiceImpl.buscarPorNombre(nombre);
}

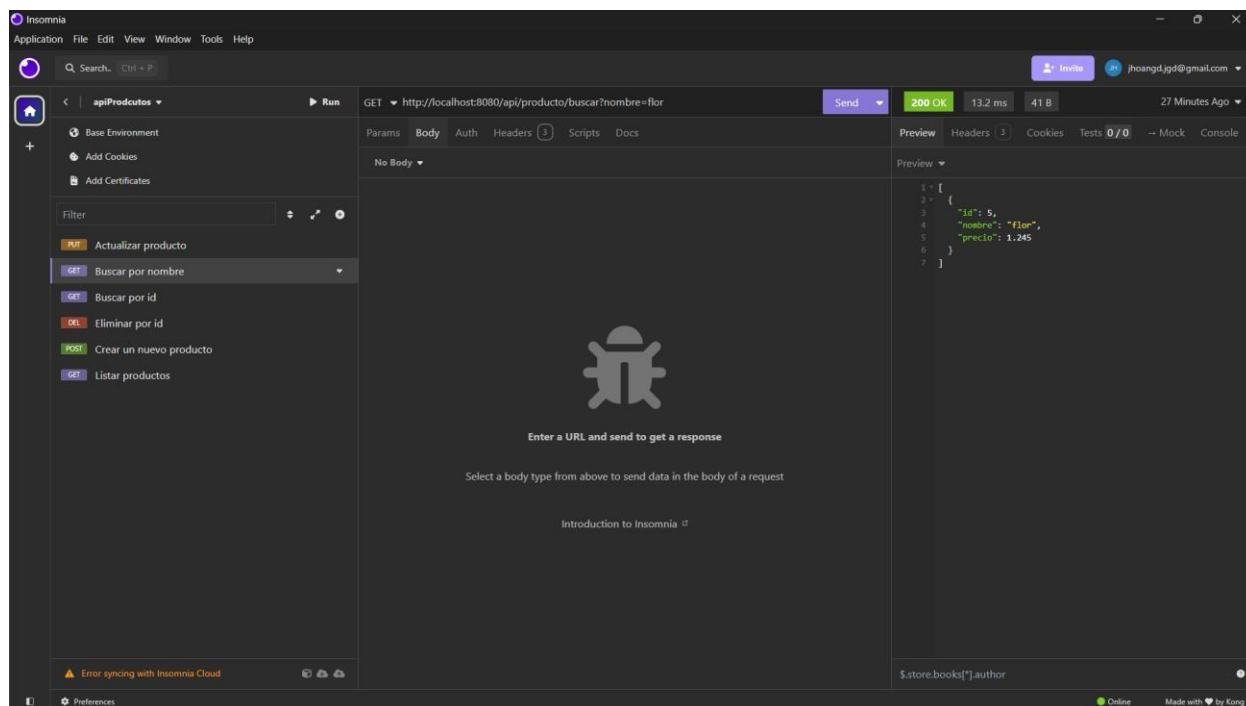
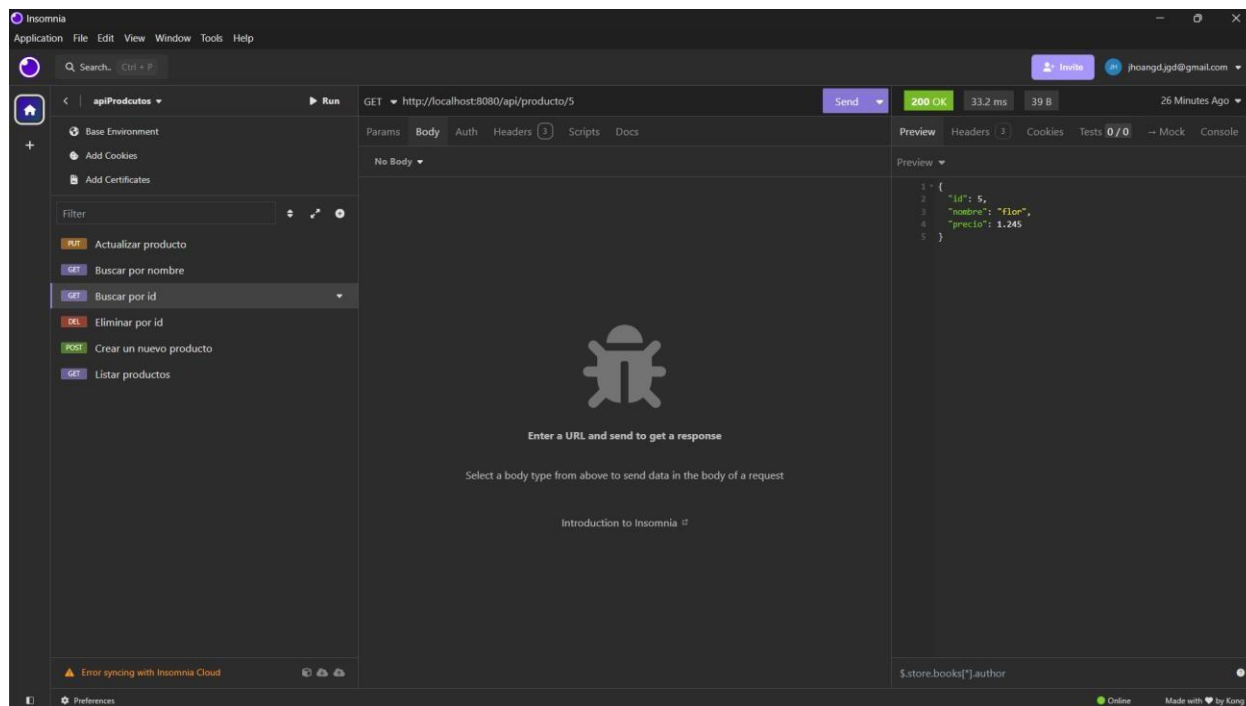
/**
 * Endpoint Recibe los datos en formato json y los guarda en memoria
 */
@PostMapping
public Producto agregarProductos(@RequestBody Producto producto) {
    productoServiceImpl.guardarProducto(producto);
    return producto;
}

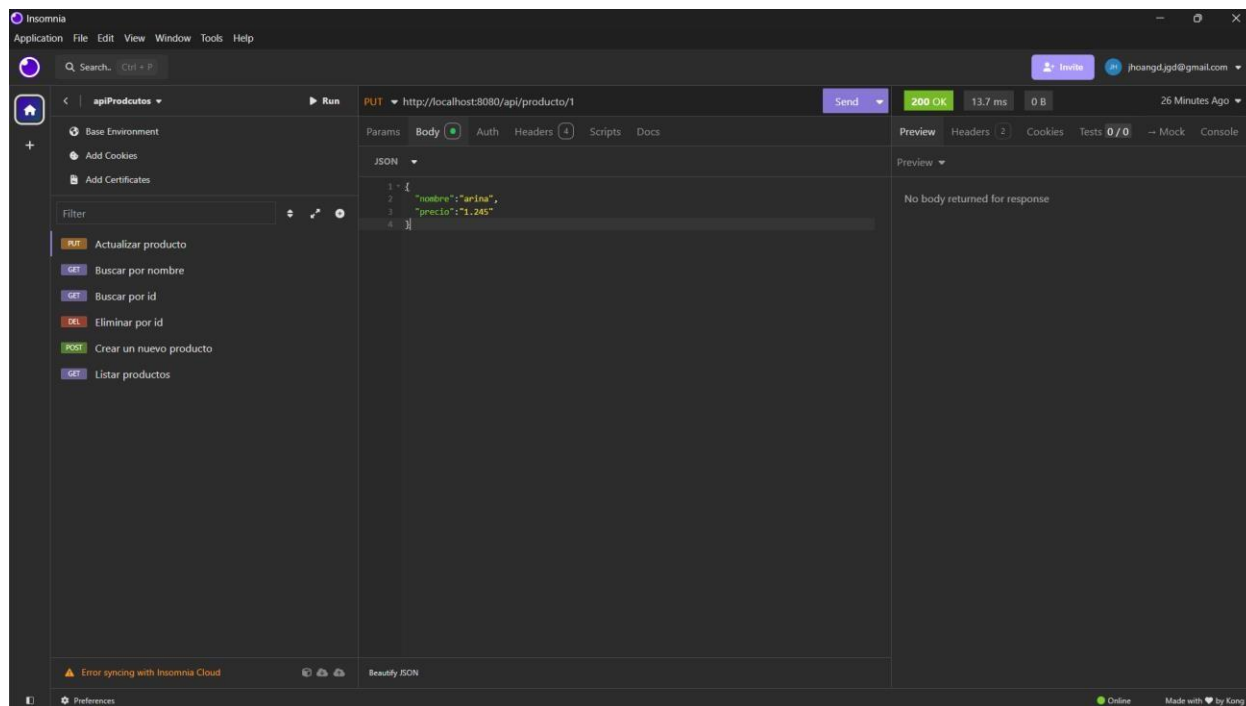
```

Pruebas de los endpoints

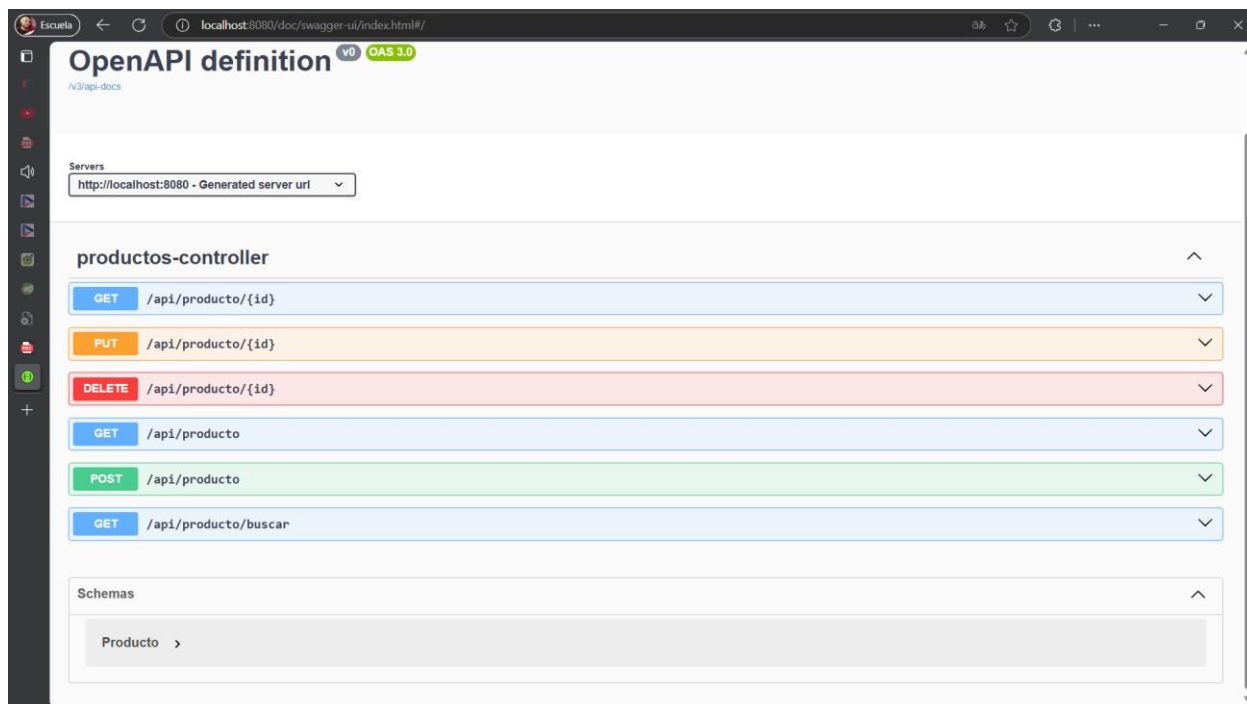








Documentacion en swagger



LINK DEL VIDEO EXPLICATIVO: <https://youtu.be/RkdtqY5BZe4?si=CcOp0aYAQW6Kaz8F>

LINK DEL PROYECTO EN GITHUD [JimsimroDev/productodApi](https://github.com/JimsimroDev/productodApi): api con crud basico en memoria no se usa aun bases de datos