**Prácticas Sesión 3.**

**Parte 1**

* Usando el Spring Initializr, generar un proyecto Spring Boot 3.2.4, para Java 17, usando maven con empaquetado JAR. Descargarlo e importarlo al IDE.
* Analizar el código generado.
* Iniciar la aplicación con alguna de las opciones:
  1. Puede usar el comando ./mvnw clean spring-boot:run o mvn clean spring-boot:run.
  2. Click derecho en el método public static void main de la clase RestapiApplication
* Agregar nuestro primer RestController.

@RestController

public class HelloController {

@GetMapping("/")

public String index() {

return "Greetings from Spring Boot!";

}

}

* Acceder al URL <http://localhost:8080/> y ver la respuesta.

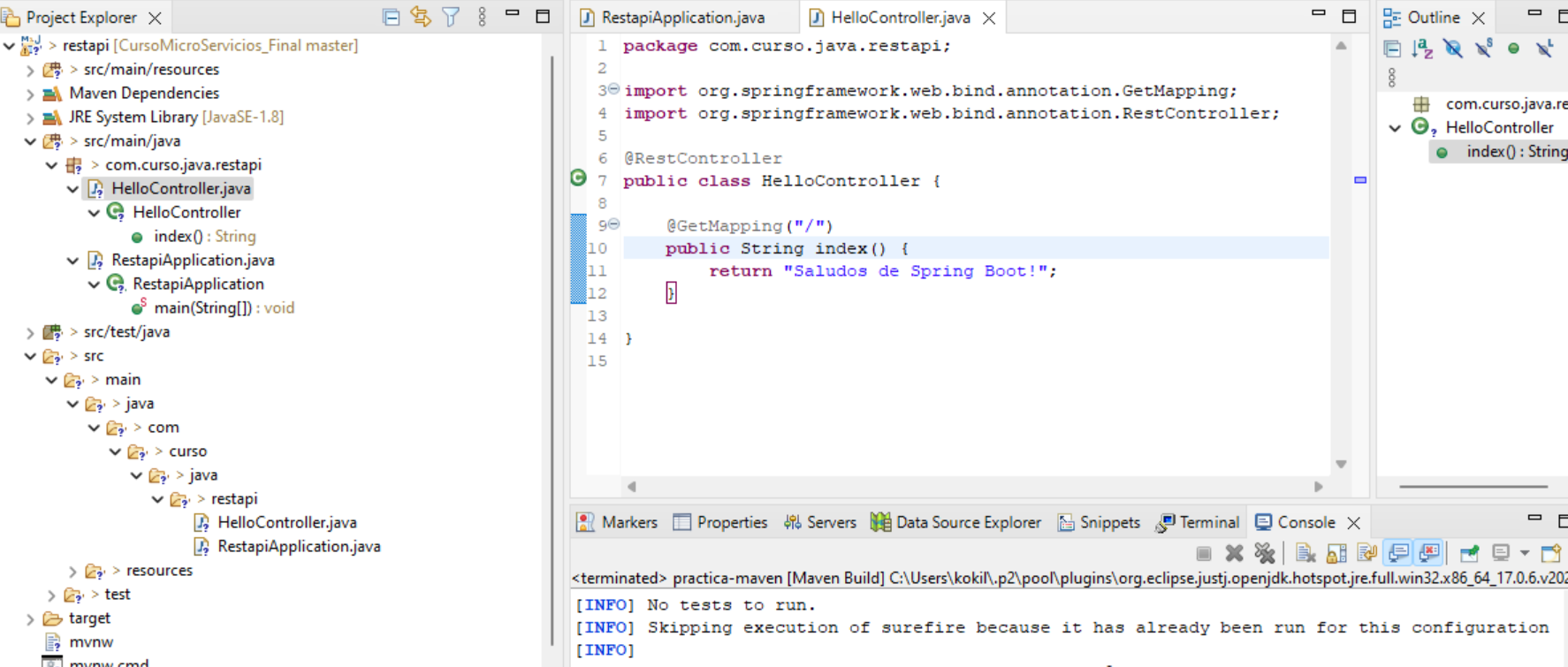
**Troubleshooting** Dependiendo de la versión de Maven, o del IDE, puede ser que tenga que agregar las siguientes lineas en el POM.xml en la sección de <properties>.

<maven.compiler.source>17</maven.compiler.source>

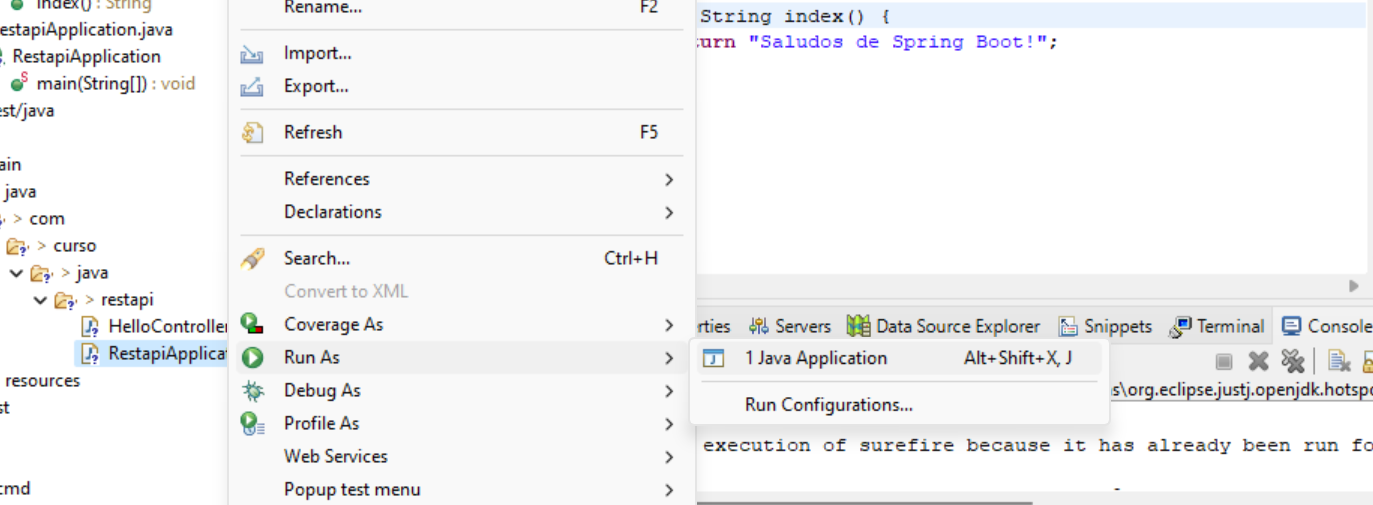
<maven.compiler.target>17</maven.compiler.target>

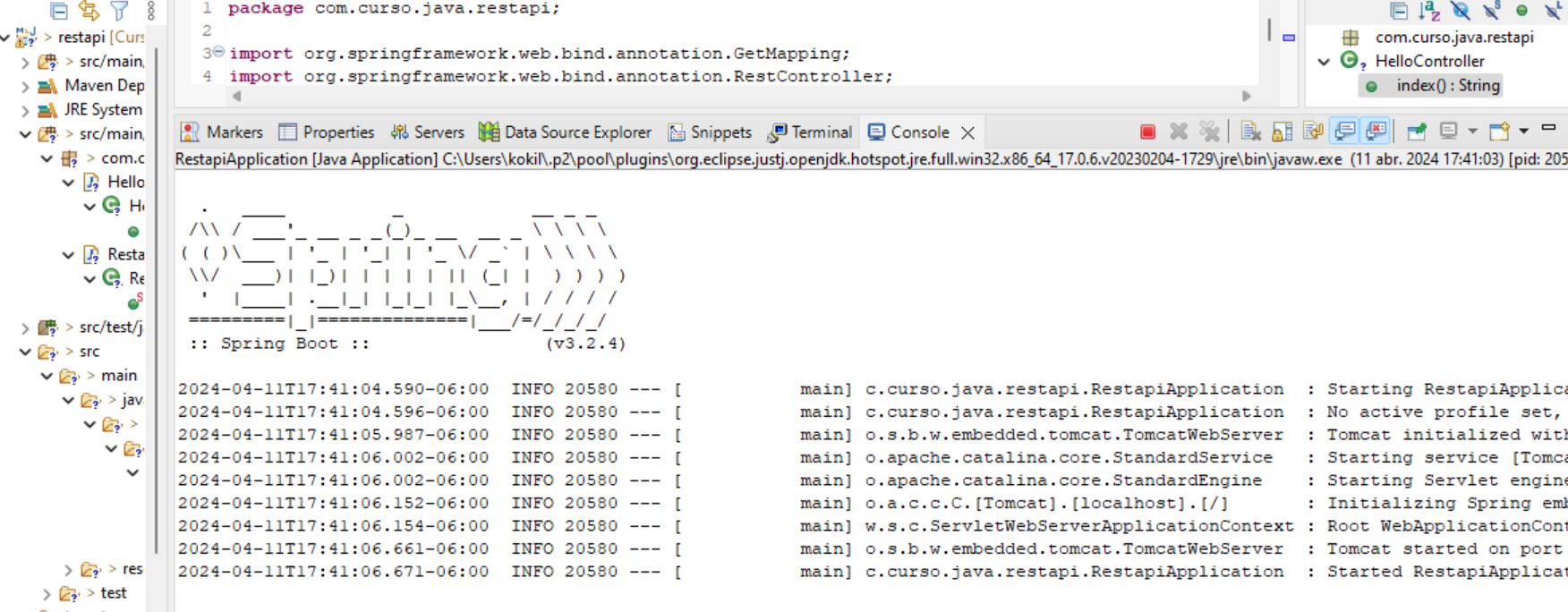
**Práctica Guíada.**

1. Tomar el Proyecto restapi indicado por el profesor en el github, abrirlo y agregarle la clase HelloController.java

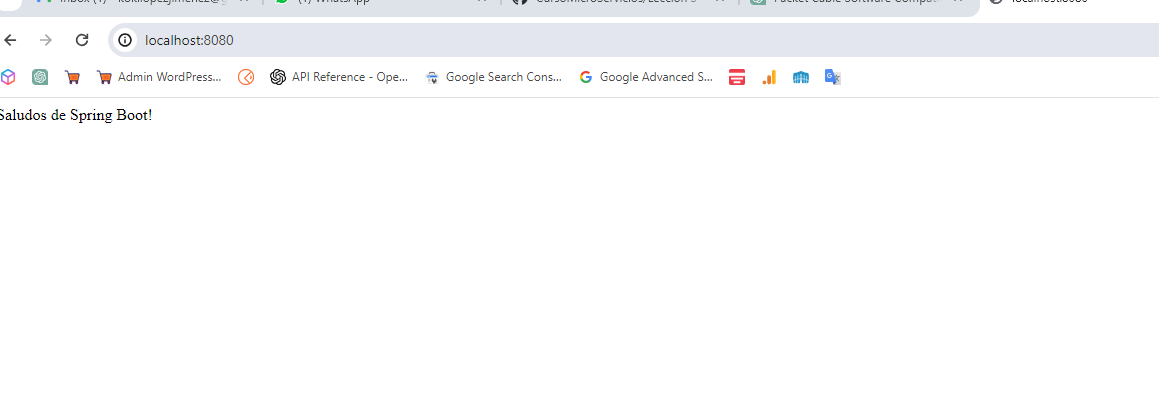


1. Según la imagen que tenga el estereotipo @RestController y la anotación @GetMapping(“/”) y retornar un String del método index “Saludfos de Spring Boot!”
2. Correr como Java Application la clase principal del proyecto RestapiApplication.java.





1. Revisar que el framework de Springboot se inicialice correctamente “Started RestApplication”.



1. Revisar en cualquier explorador de internet que mientras el framework de Springboot esté corriendo en la dirección de localhost:8080 la aplicación responda “Saludos de Springboot”.
2. **Final.**

**Parte 2**

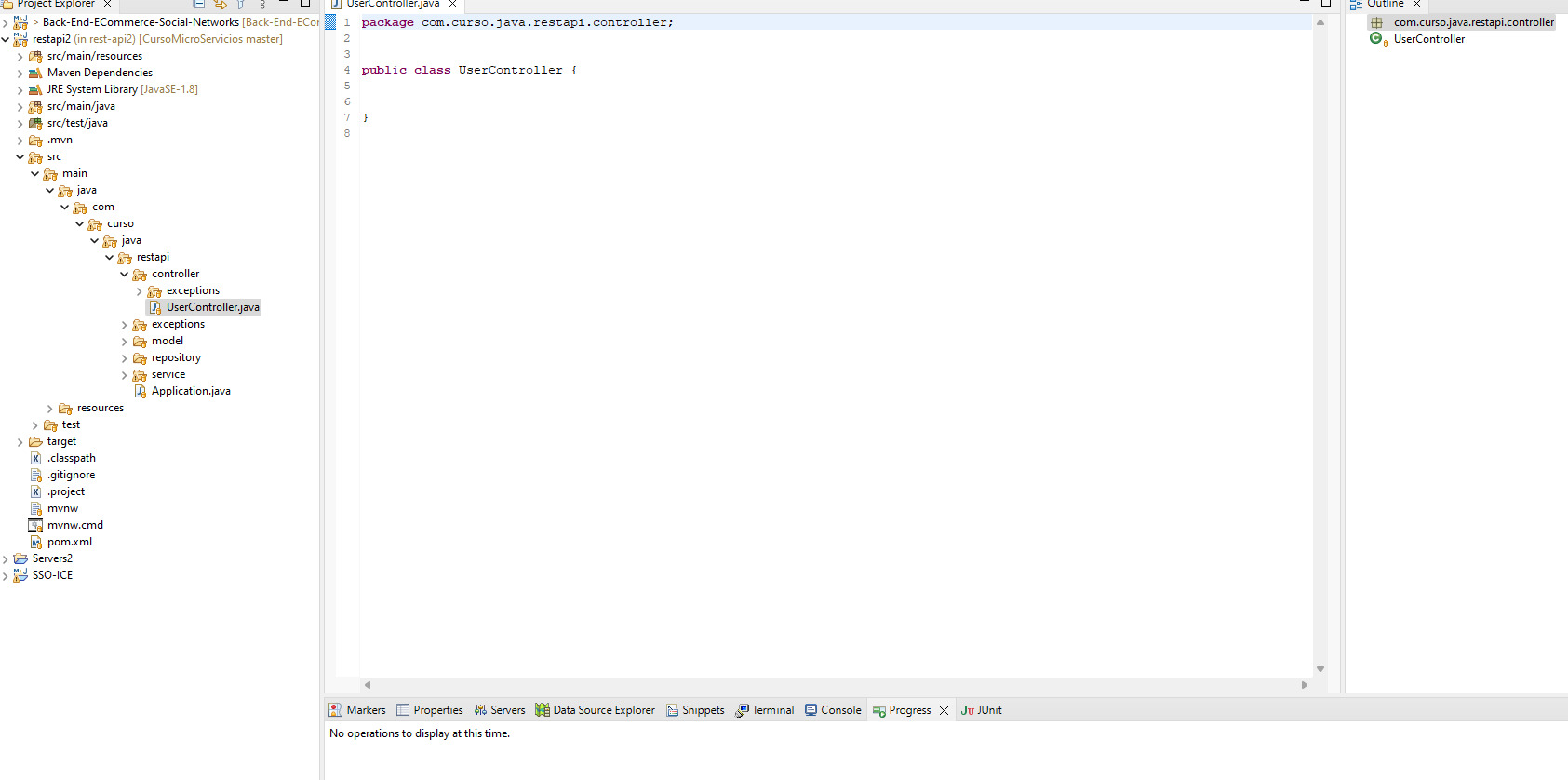
Usando como base el proyecto rest-api2.

1. Crear un rest controller que permita realizar operaciones CRUD (consultar, crear, modificar, eliminar) para usuarios.
   1. Los endpoints debe estar en la ruta /api/users
   2. Utilizar el Service existente en el proyecto
   3. Utilizar Postman para consumir los endpoints
   4. Manejar las excepciones para validaciones o eventos

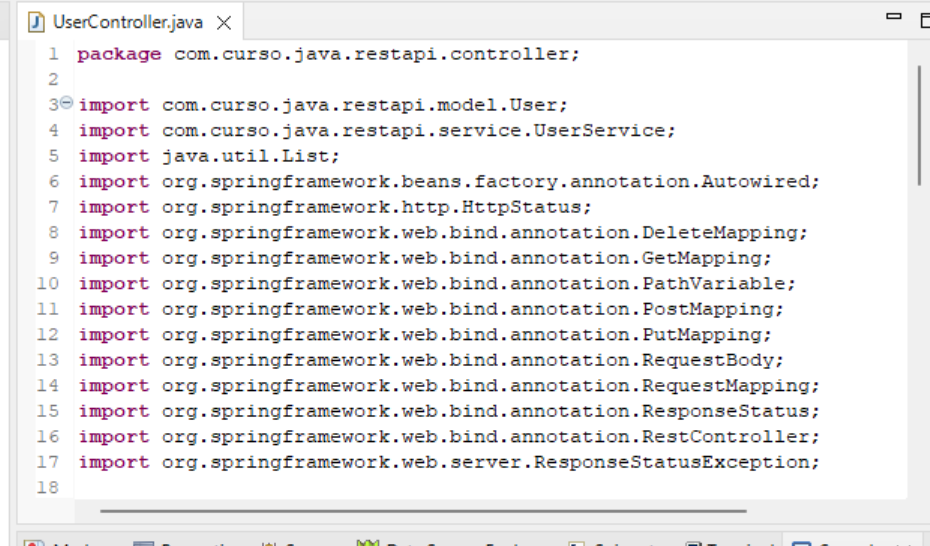
Puede utilizar Postman, e importar la colleción creada para el laboratorio con los llamados a ejecutar para realizar las pruebas al API.

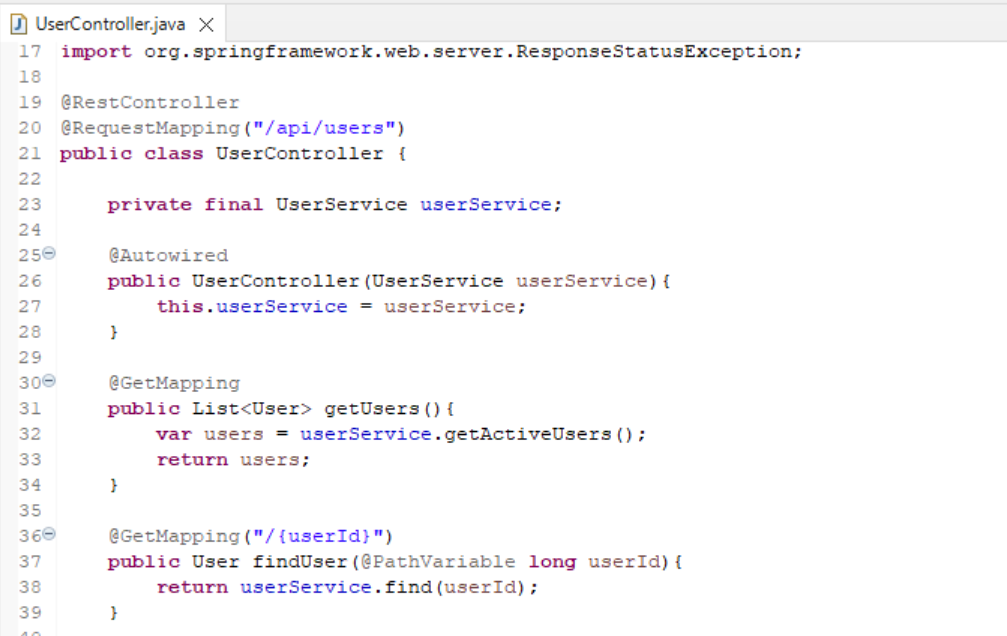
**Práctica Guíada.**

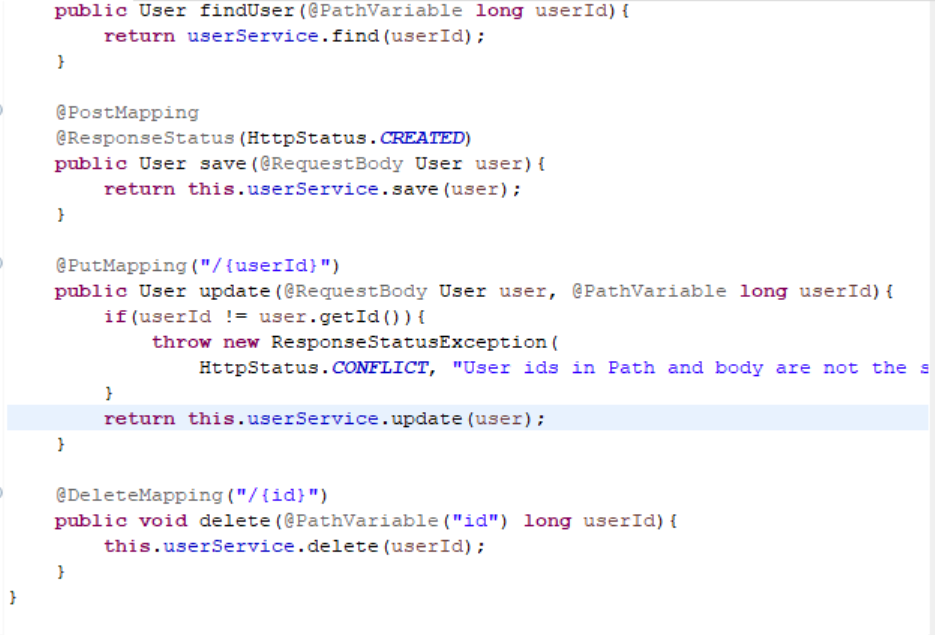
* 1. Utilizar la clase creada UserController.java en la carpeta controller y paquete controller.

****

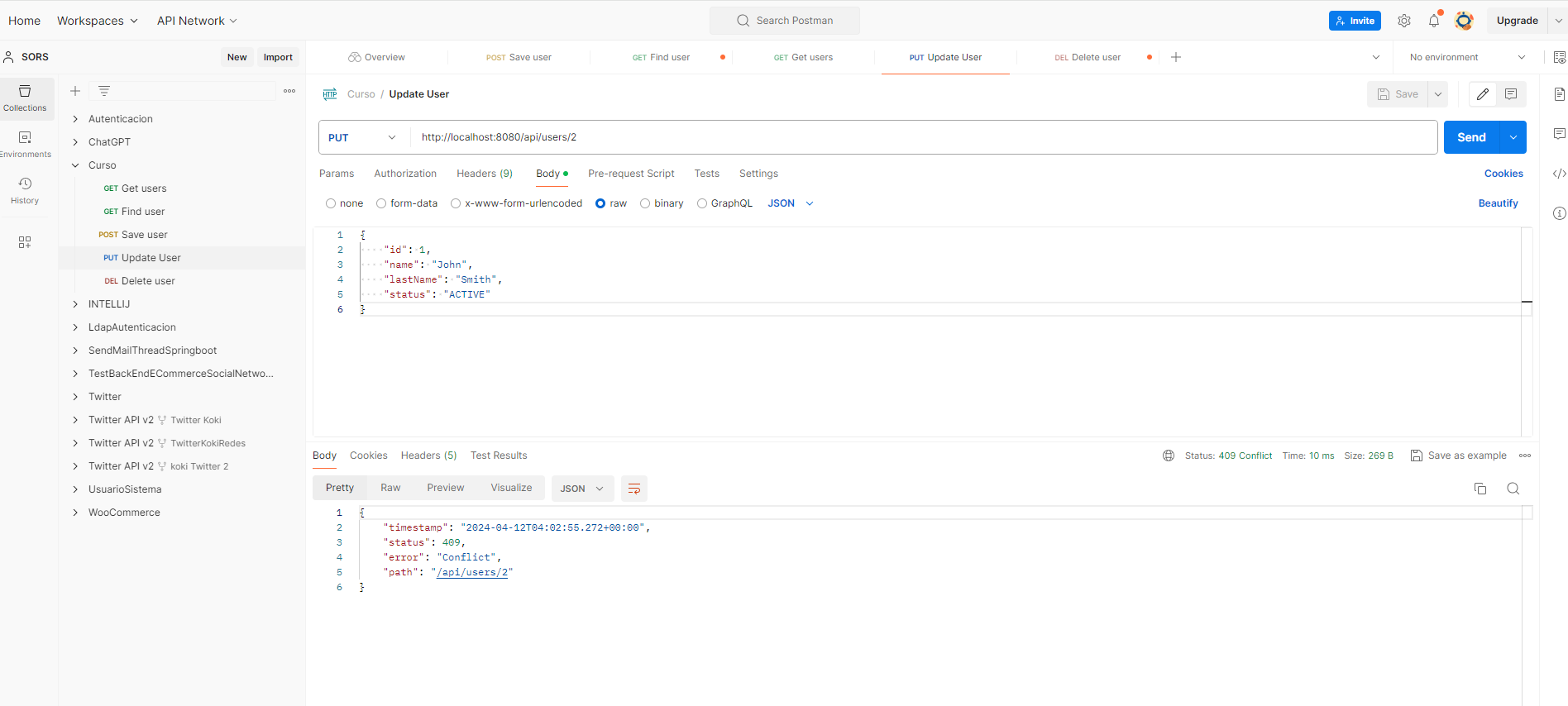
Debe importar las siguientes de las siguientes librerías para el controlador.



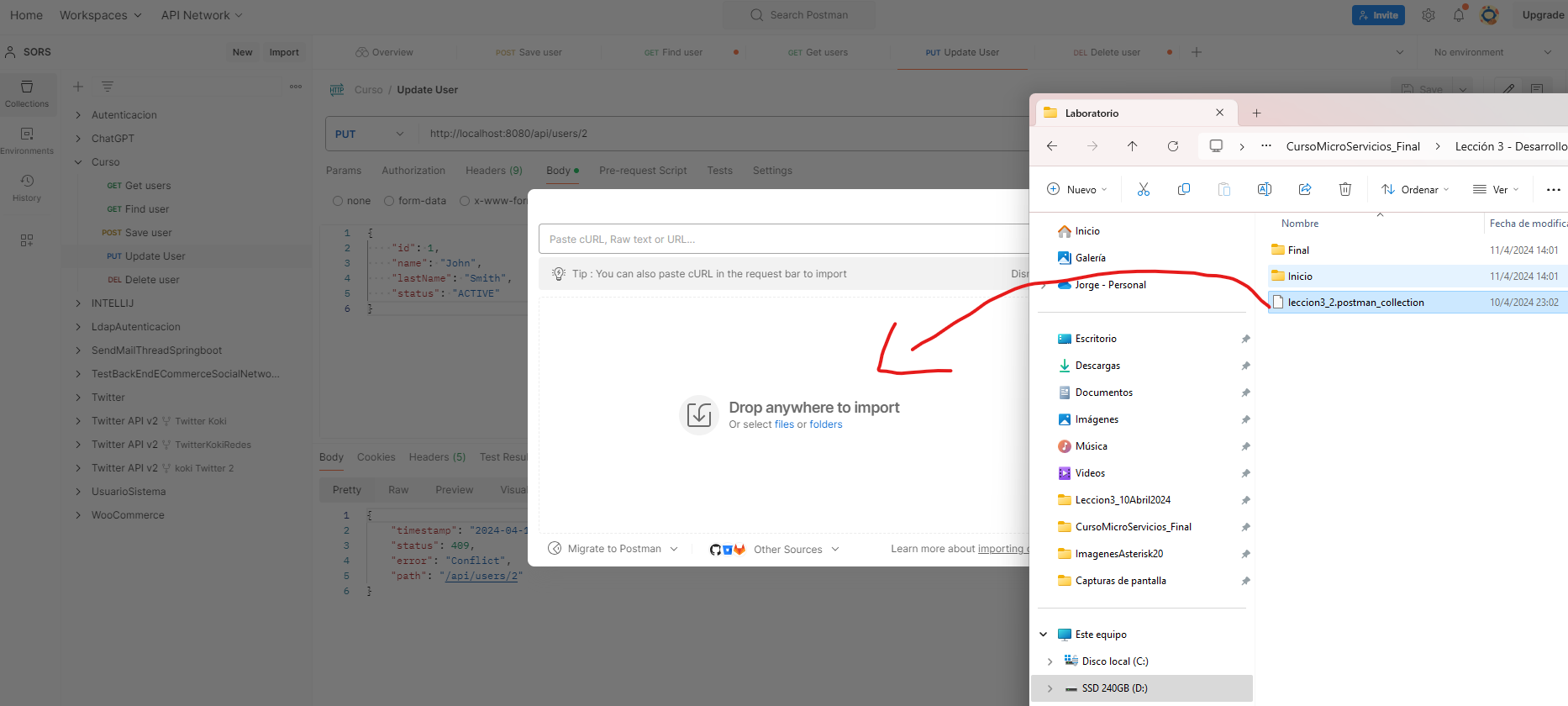
Debe crear las operaciones CRUD con las respectivas operaciones básicas de Rest Html básicas POST, GET, PUST, DELETE. Inyección de dependencia UserService por constructor, recordar hacer el listar todos los usuarios como el obtener el usuario por su id y manejar las excepciones, use dos figuras siguientes.



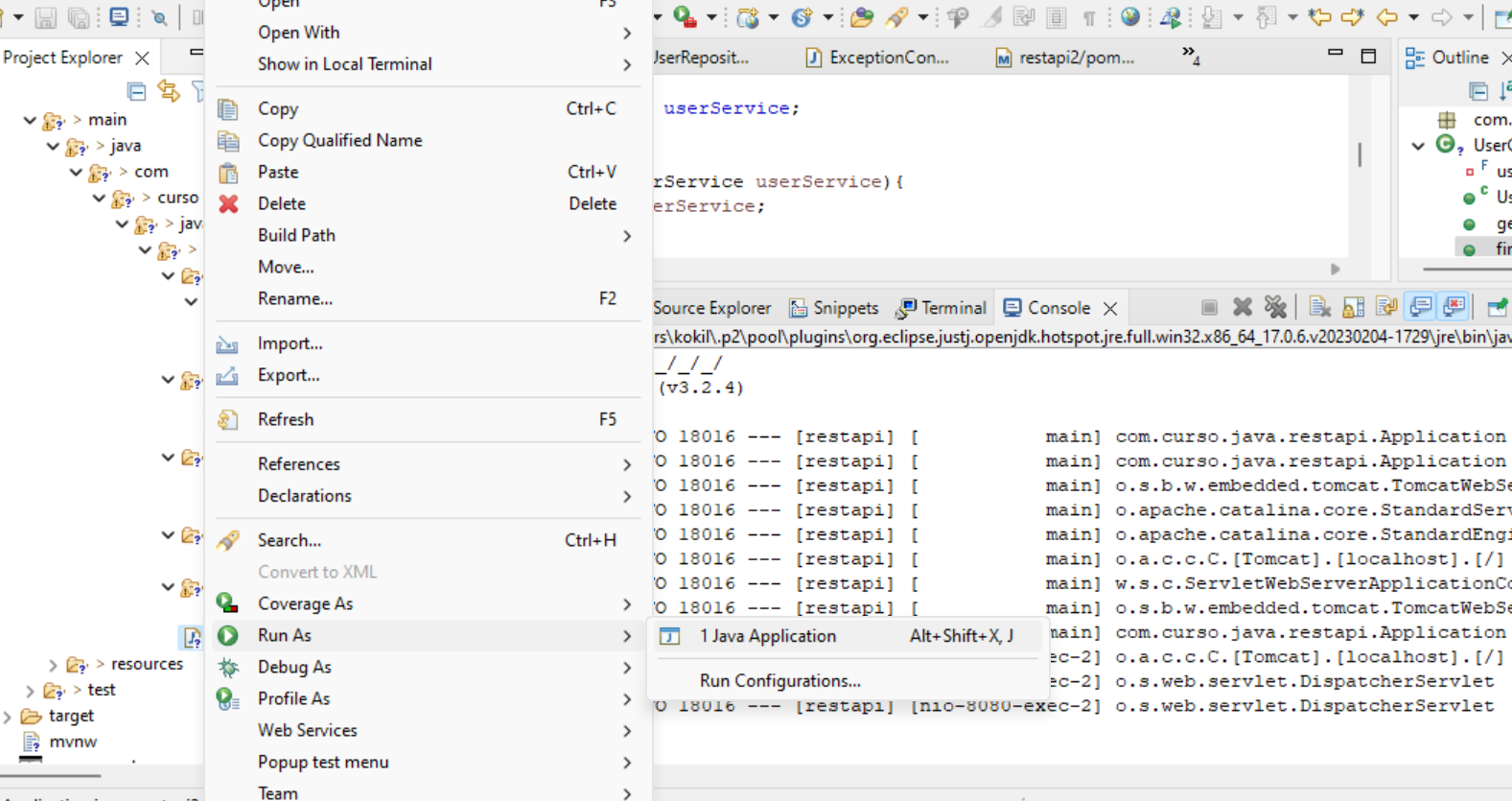
Abrir Postman ya sea con su usuario o registrarse como anónimo o con un usuario con su email.



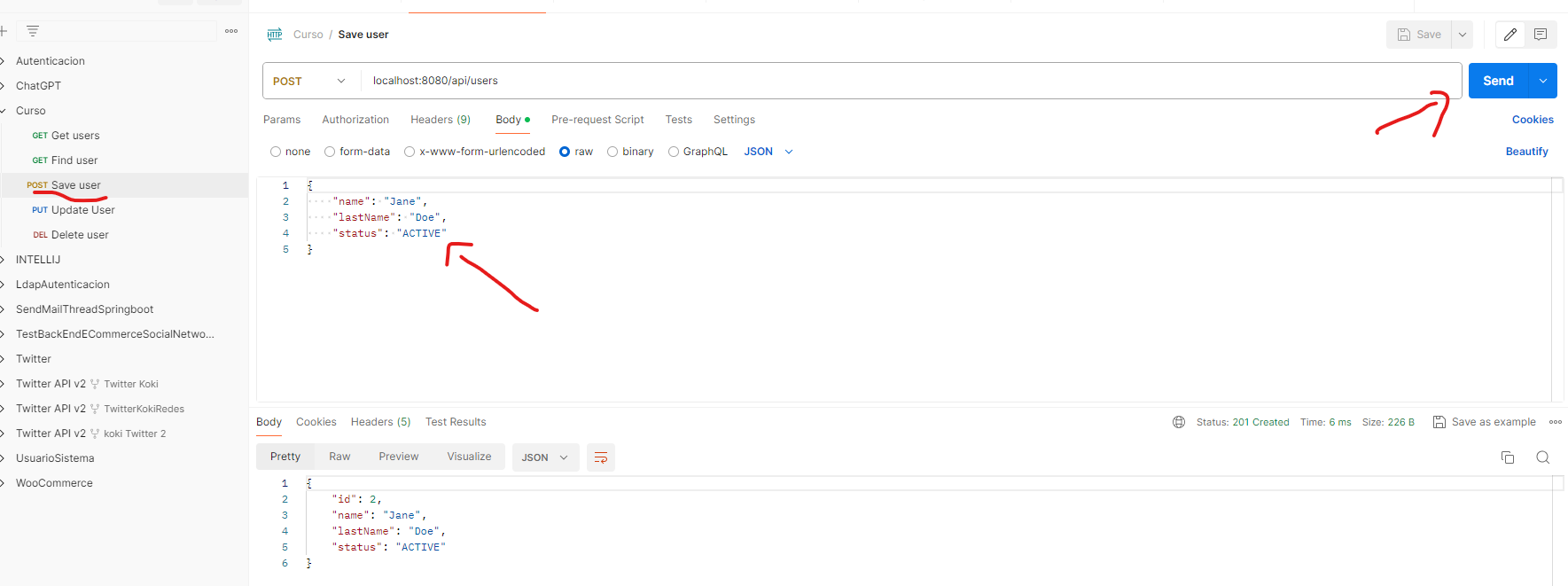
Agregue la colleción de postman que viene en el zip de la práctica guíada que bajo de Moodle o de la librería de Git en el directorio de restapi2.



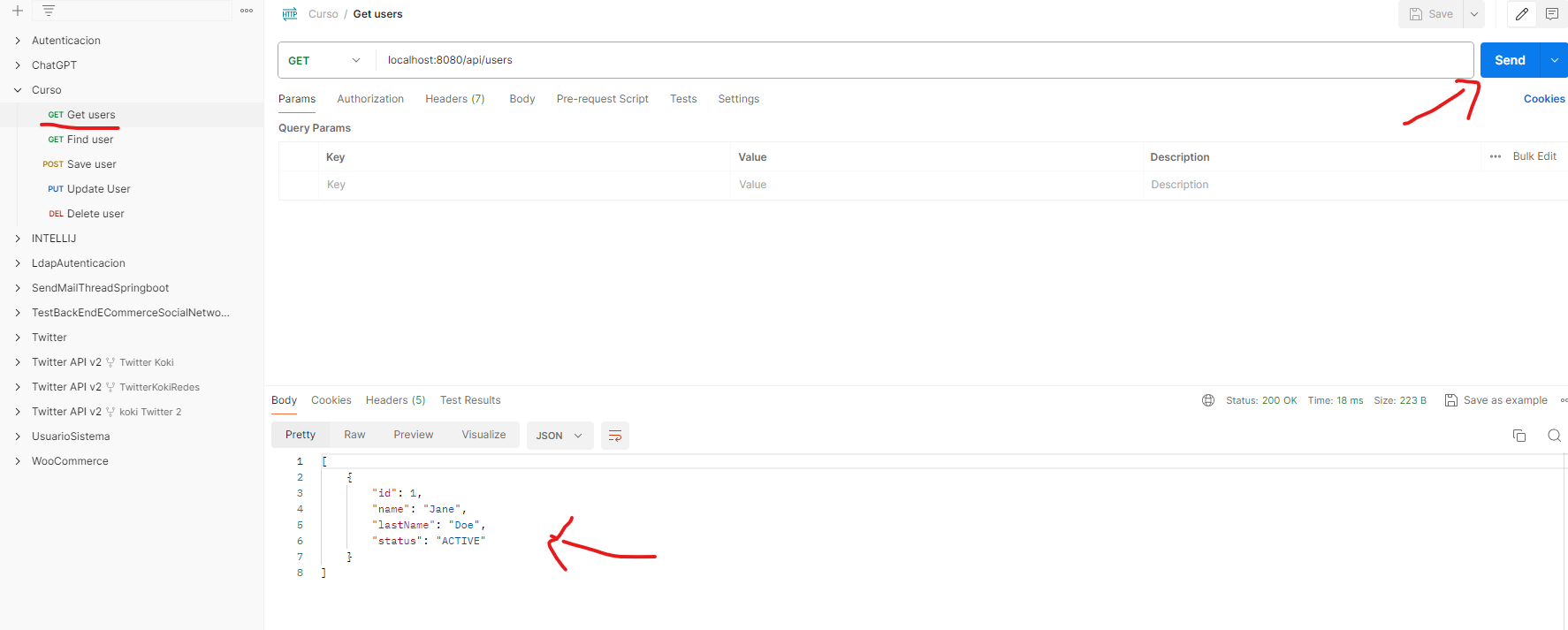
Correr la aplicación restapiApplication que es la principal como una Java Application.



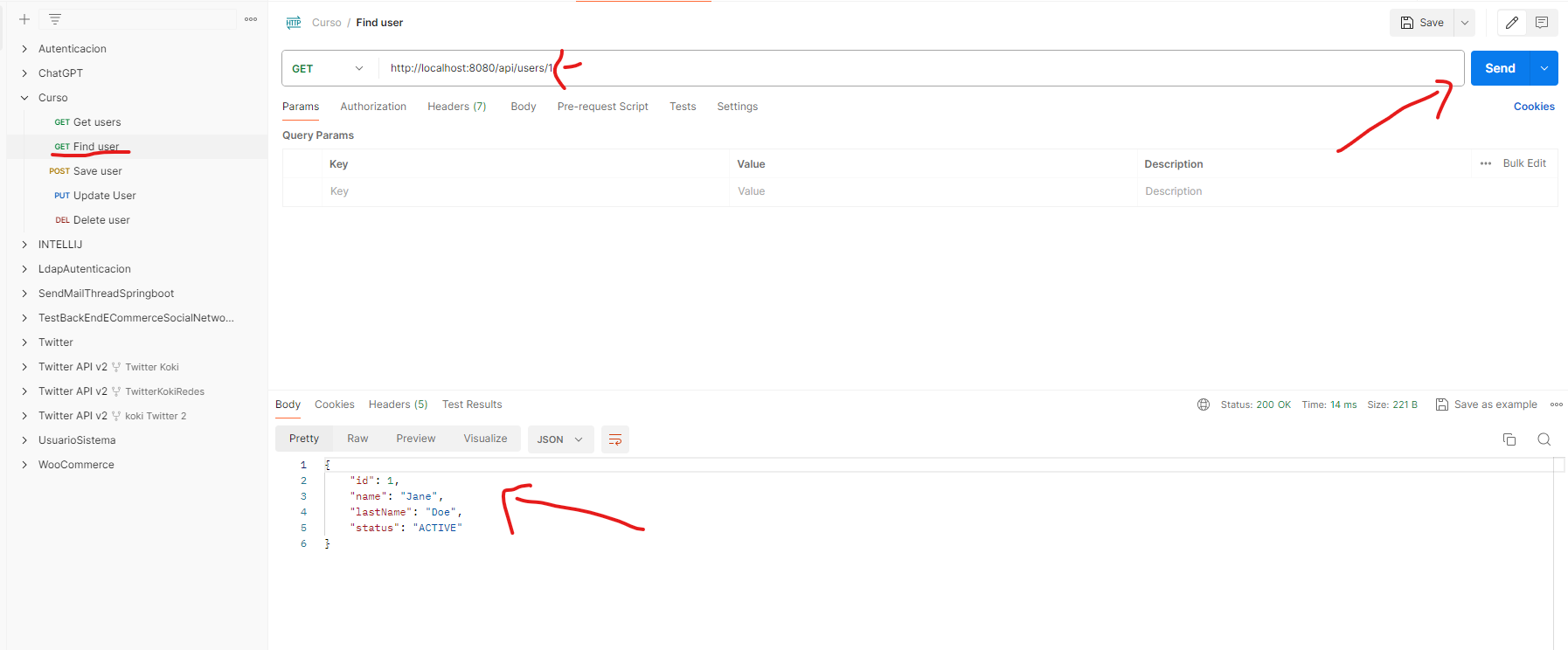
Crear un usuario nuevo.



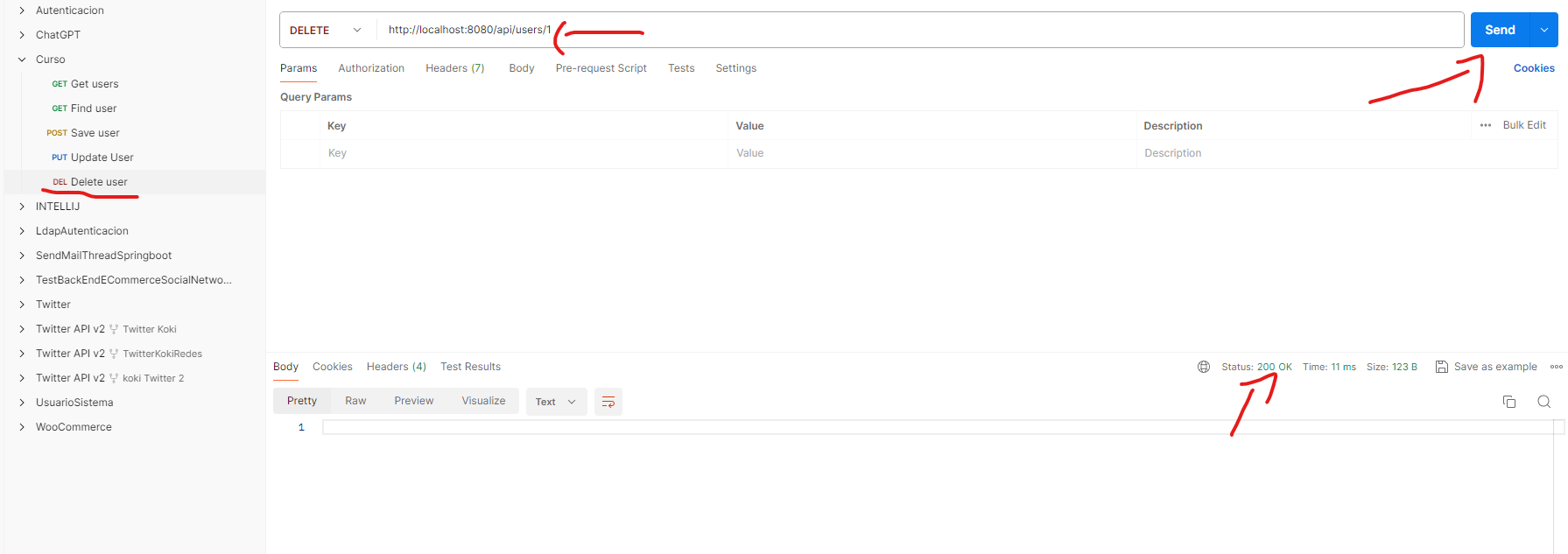
Listar los usuarios.



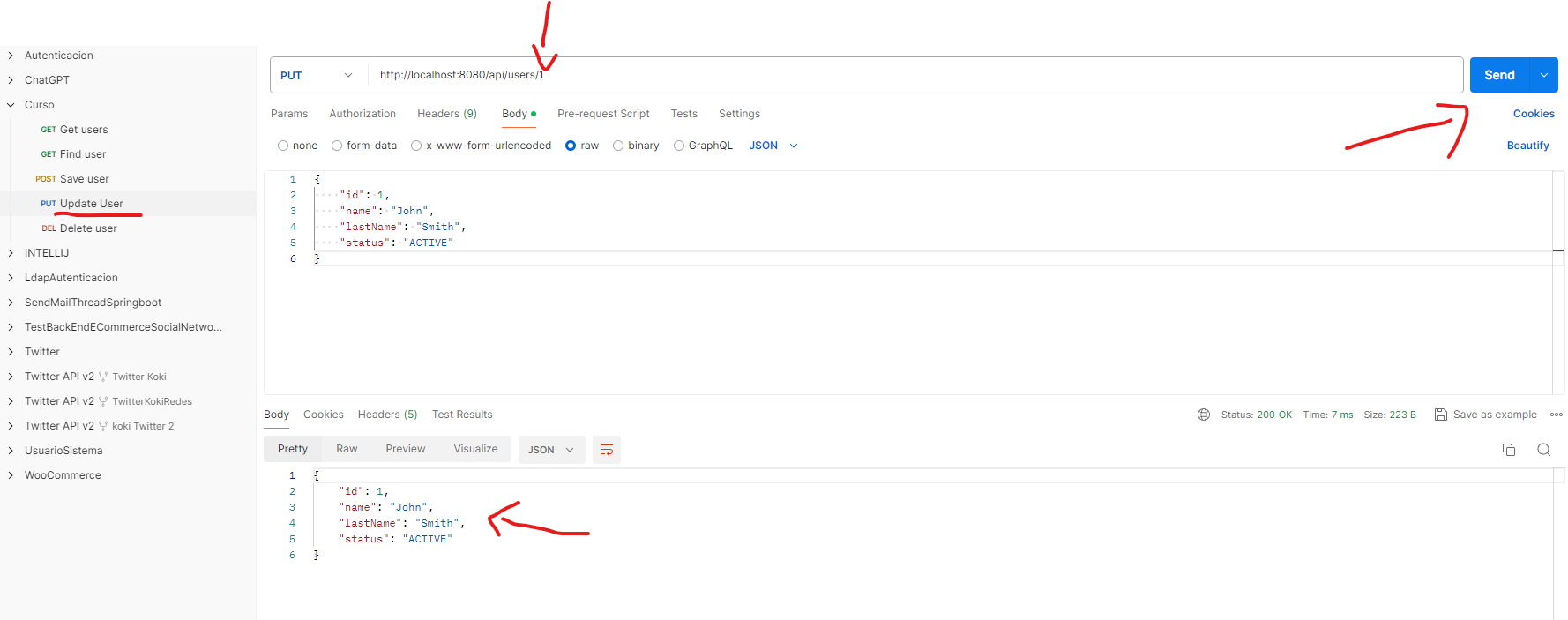
Buscar un Usuario por id.



Borrar Usuario por id.



Modificar usuario por id.



**Final.**