

Práctica RESTful

SISTEMAS ORIENTADOS A SERVICIOS

DLSIIS

Vidal Peña, Arturo
Melero Deza, Javier

22 de abril de 2019

Índice de contenidos

| | | |
|---|---|----|
| 1 | Resumen del diseño del servicio | 1 |
| 2 | Capturas de la ejecución en un cliente REST (Postman) | 2 |
| 3 | ANEXO: Problemas encontrados | 14 |

1. Resumen del diseño del servicio

La arquitectura de la aplicación consta de una base de datos simple formada por 4 tablas:

- Usuarios
- Mensajes
- Privados
- Amistad

Posteriormente se programan los recursos, modelos y servicios necesarios para cumplir con las especificaciones pedidas utilizando Jersey Restful.

Dicho servidor se ejecuta sobre Apache Maven en *localhost* con puerto 8080. Internamente, la aplicación proporciona servicios desde la clase `UserResource` utilizando el resto de clases con las estructuras adecuadas para la conexión con la base de datos y posteriormente su conversión a JSON o XML.

2. Capturas de la ejecución en un cliente REST (Postman)

Hay que poner detalles tanto de la invocación como del resultado de la operación.

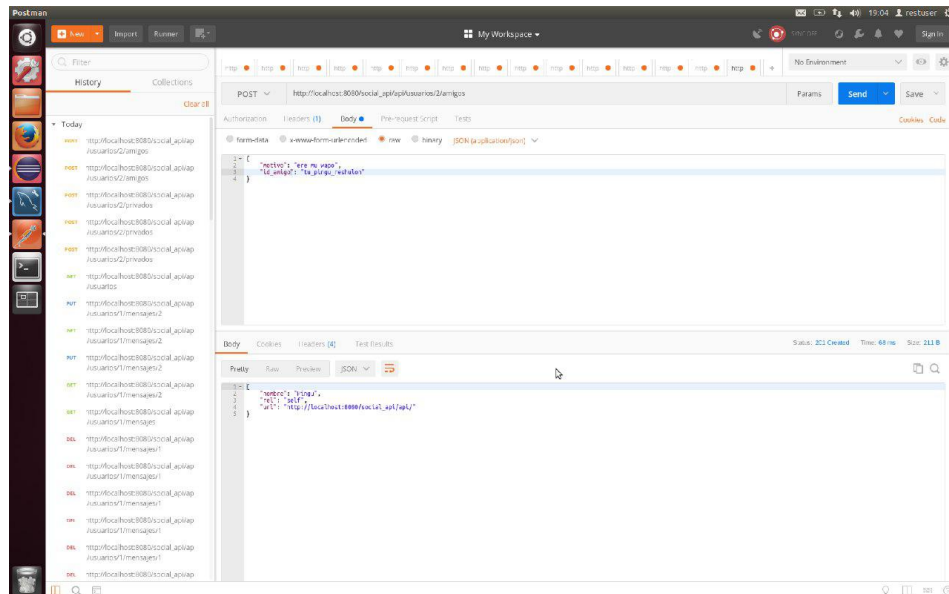


Figura 1: Método POST para insertar un amigo en el usuario con id 2.

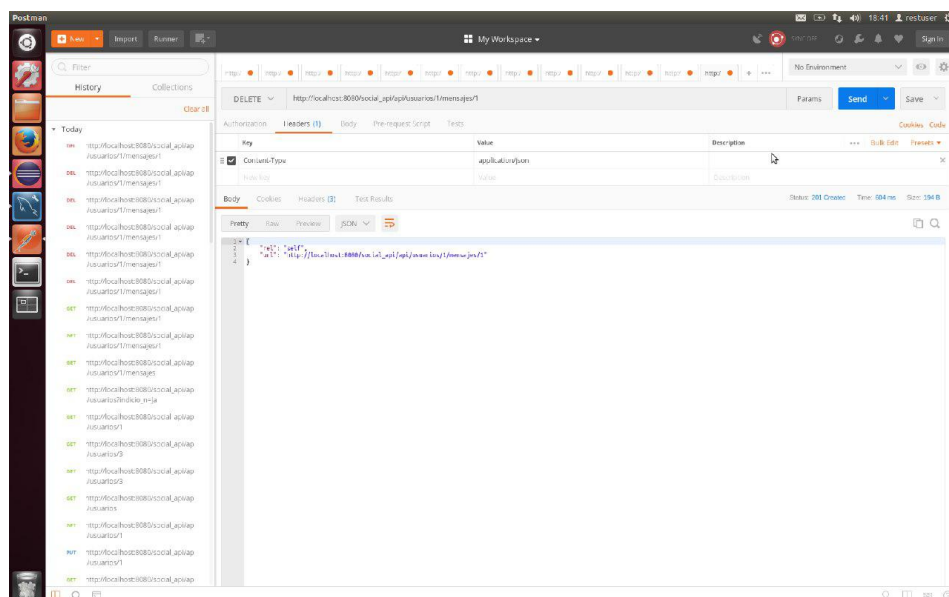


Figura 2: Método DELETE para borrar un mensaje del usuario con id 1.

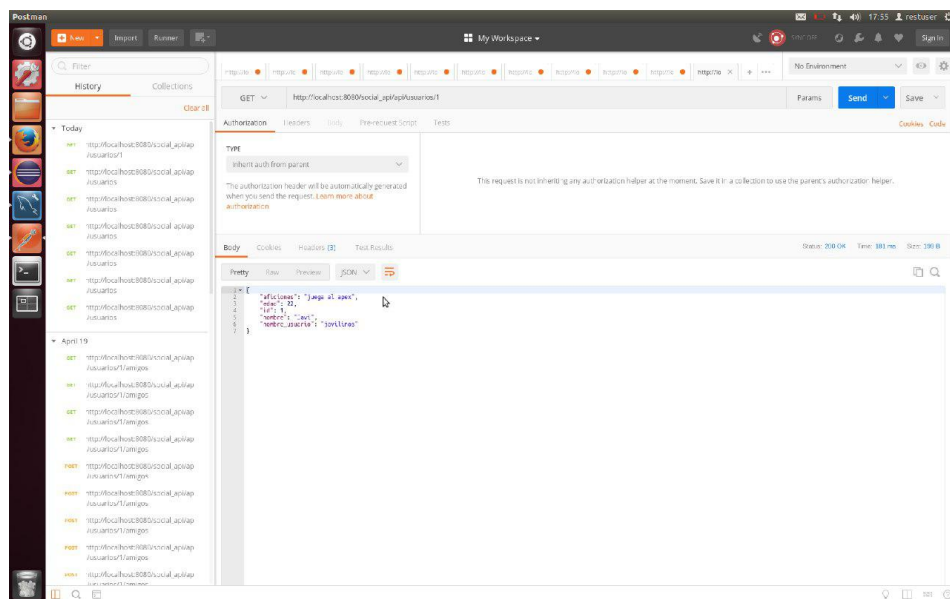


Figura 3: Método GET para obtener la información del usuario con id 1.

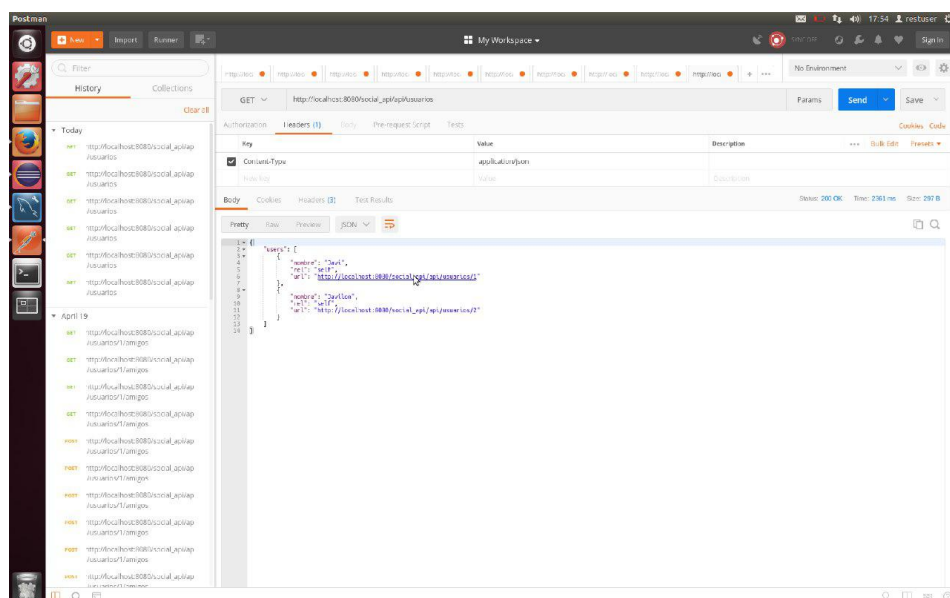


Figura 4: Método GET para obtener la información de todos los usuarios.

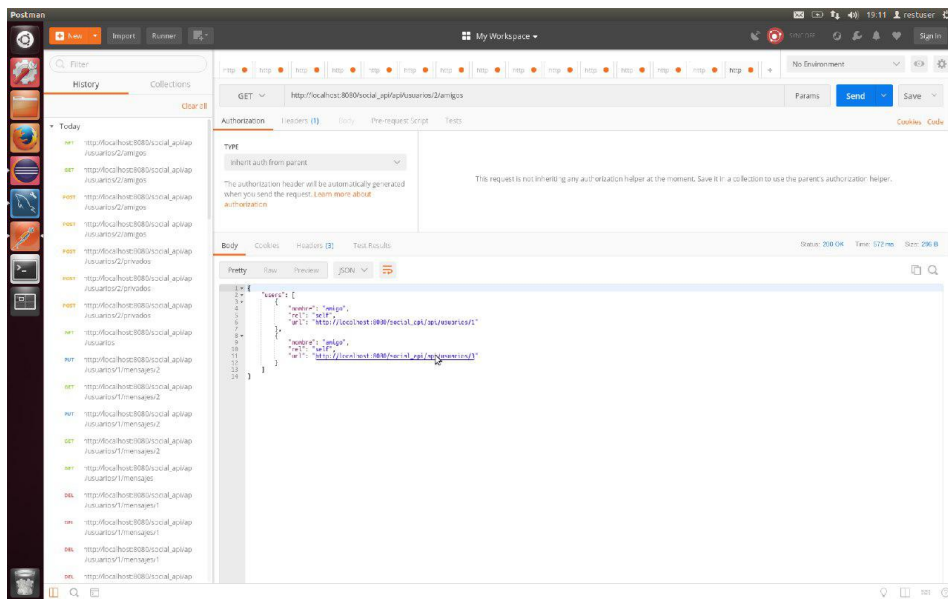


Figura 5: Método GET para obtener los amigos del usuario con id 2.

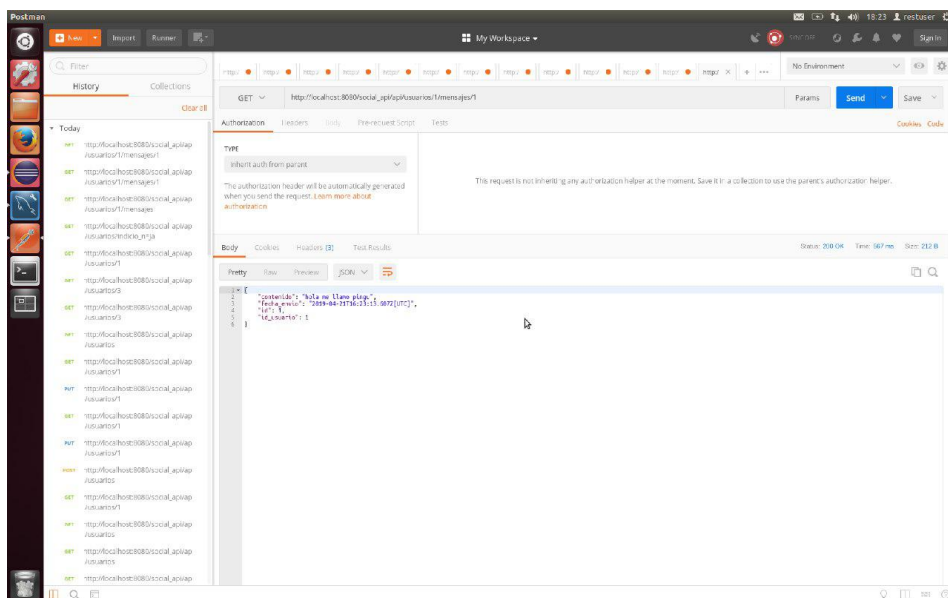


Figura 6: Método GET para obtener la información del mensaje con id 1 del usuario con id 1.

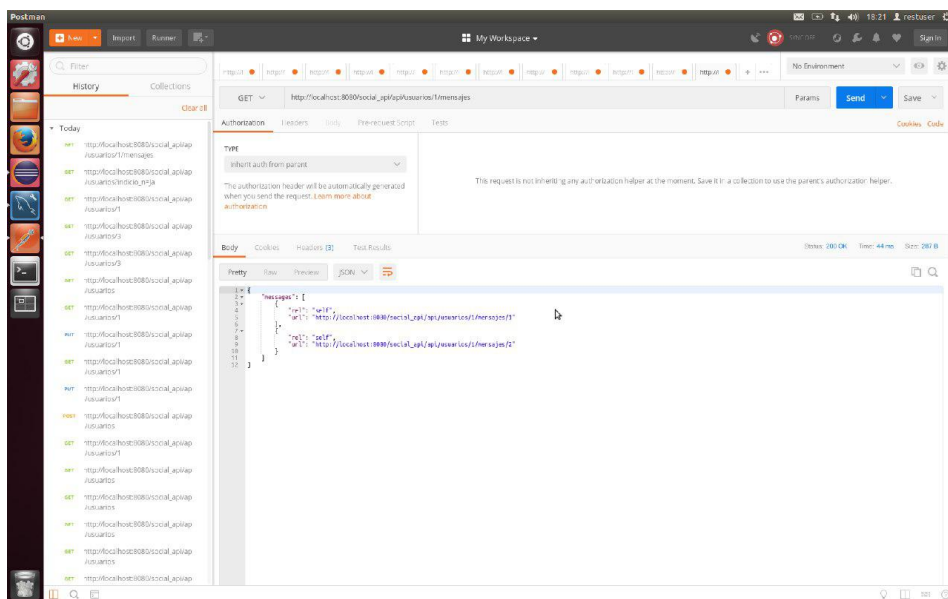


Figura 7: Método GET para obtener los mensajes del usuario con id 1.

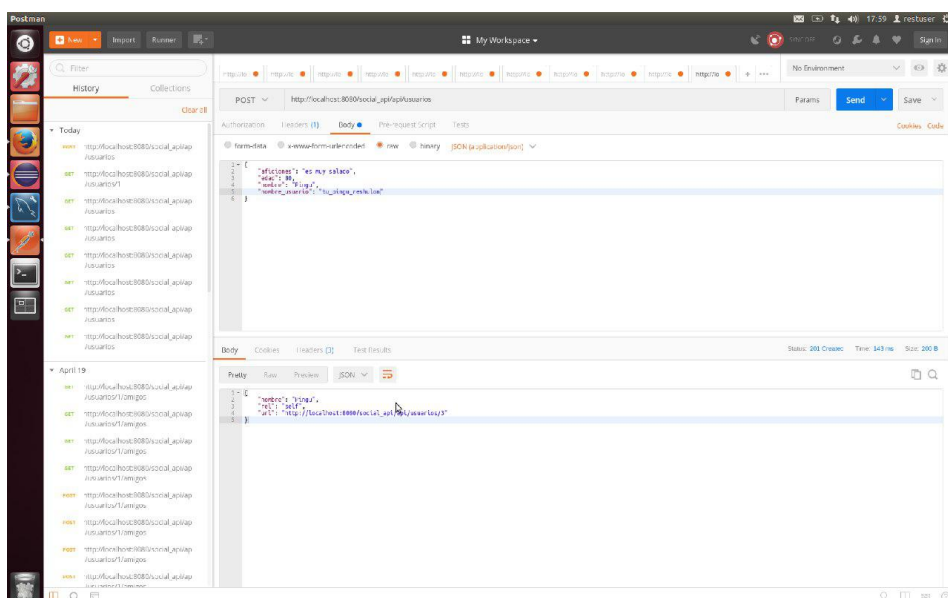


Figura 8: Método POST para añadir un nuevo usuario con id 3.

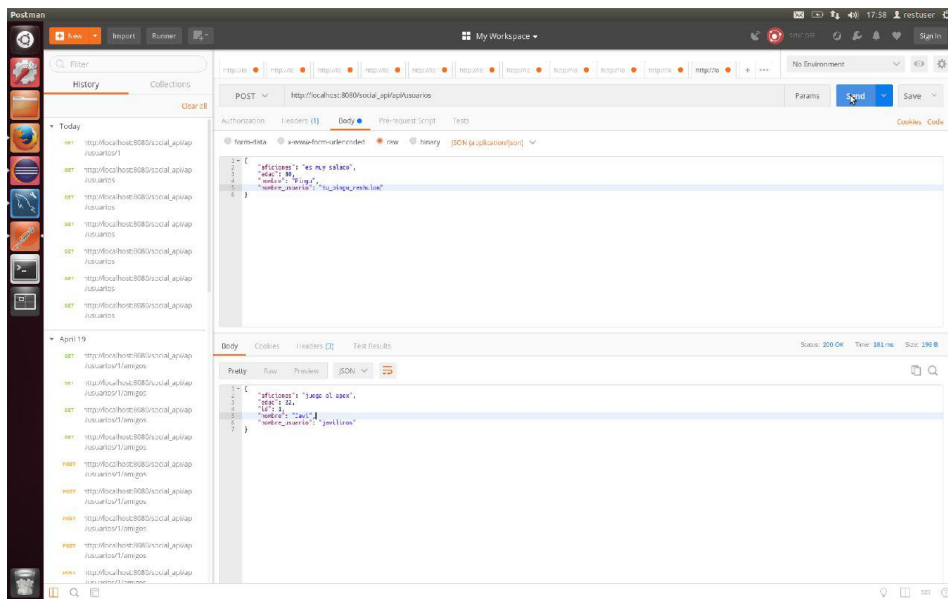


Figura 9: Método POST para añadir un nuevo usuario con id 1.

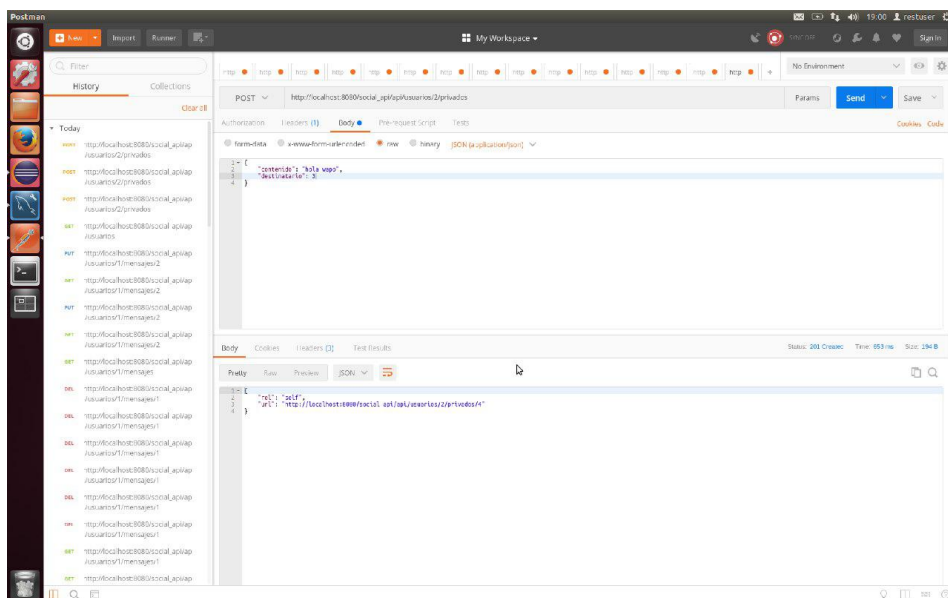


Figura 10: Método POST para añadir un nuevo mensaje privado desde el usuario con id 2 al usuario con id 3.

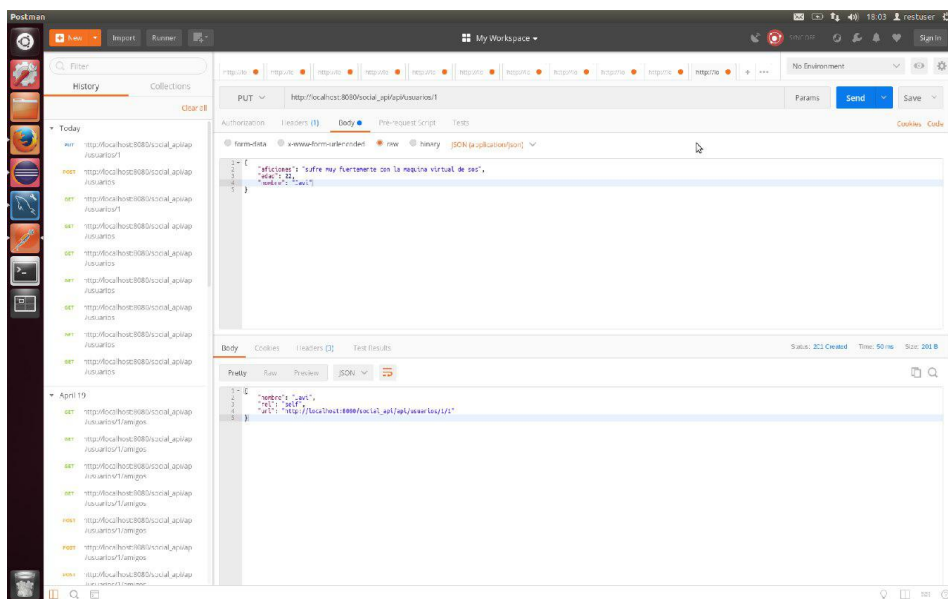


Figura 11: Método PUT para modificar datos pertenecientes al usuario con id 1.

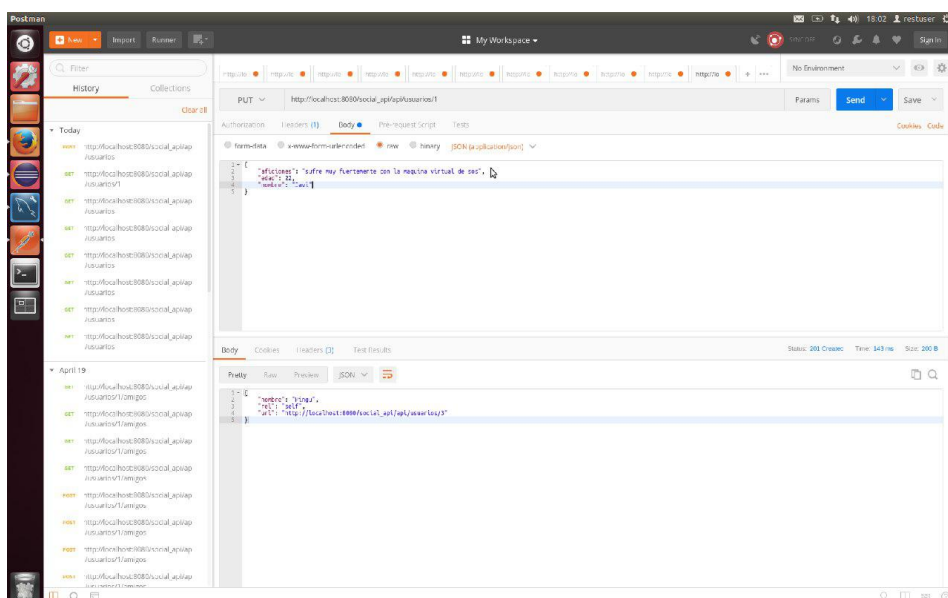


Figura 12: Método PUT para modificar datos pertenecientes al usuario con id 1.

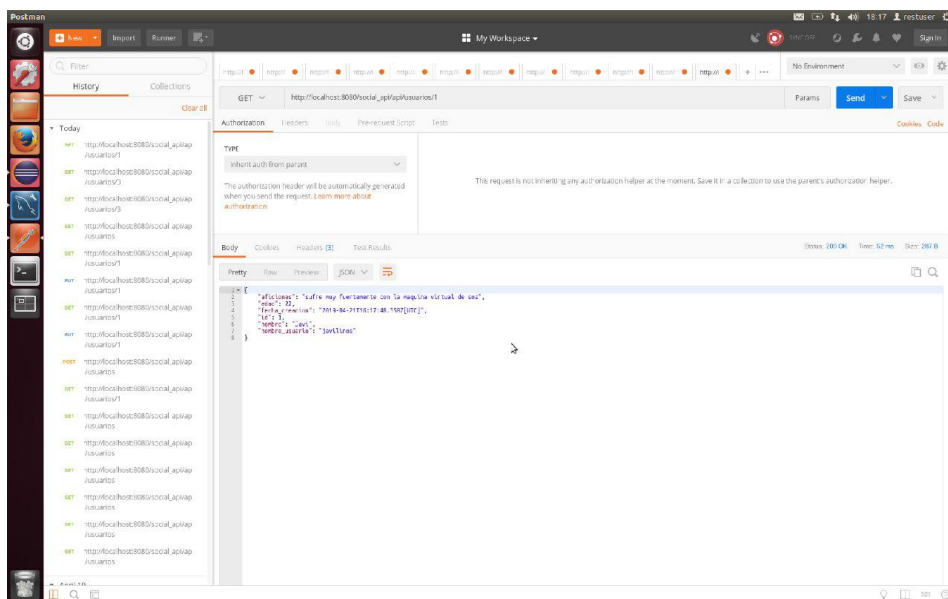


Figura 13: Método GET para obtener la información del usuario con id 1.

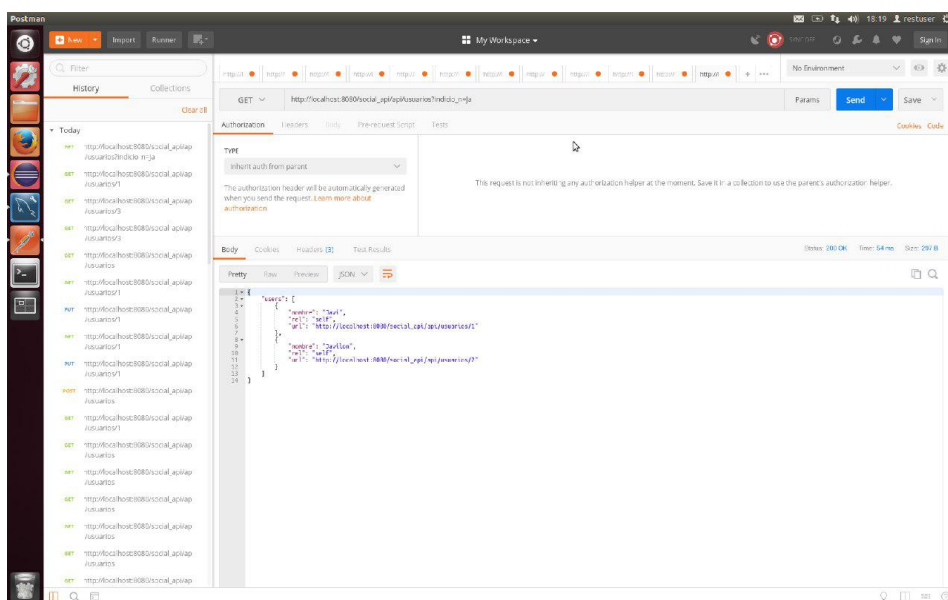


Figura 14: Método GET para obtener los usuarios cuyo nombre empieza por "Ja".

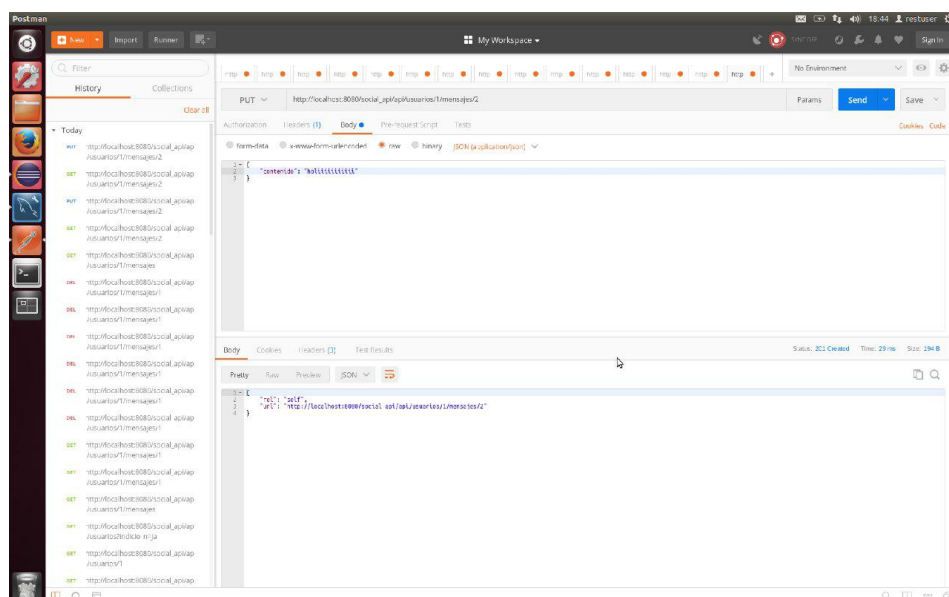


Figura 15: Método PUT para modificar el mensaje con id 2 de la página del usuario con id 1.

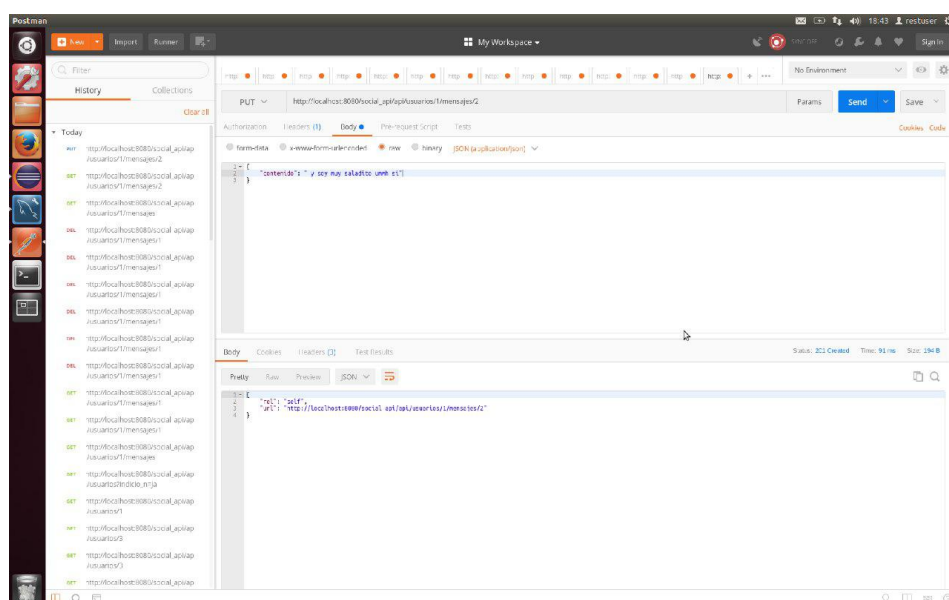


Figura 16: Método PUT para modificar el mensaje con id 2 de la página del usuario con id 1.

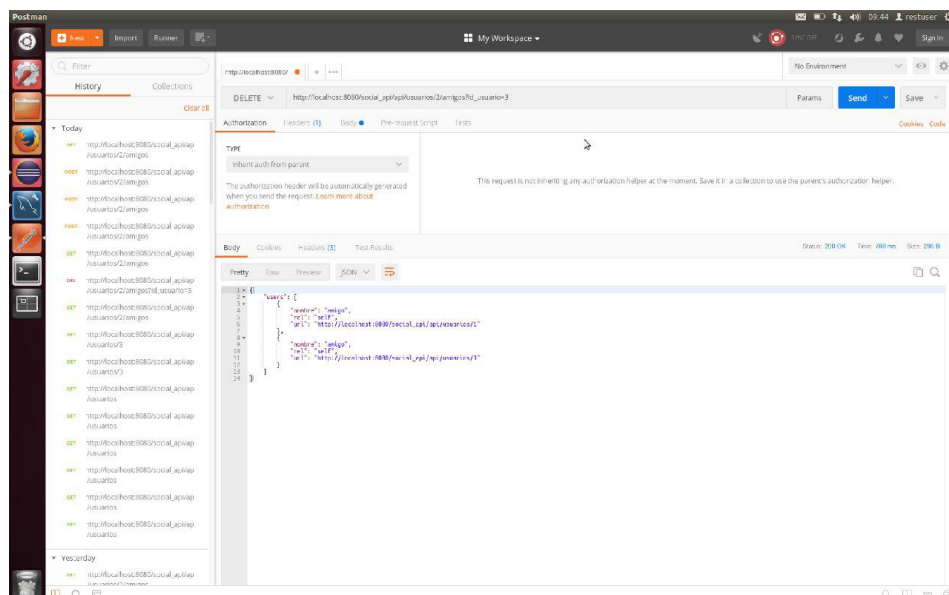


Figura 17: Método DELETE para eliminar al usuario con id 3 de la lista de amigos del usuario con id 2.

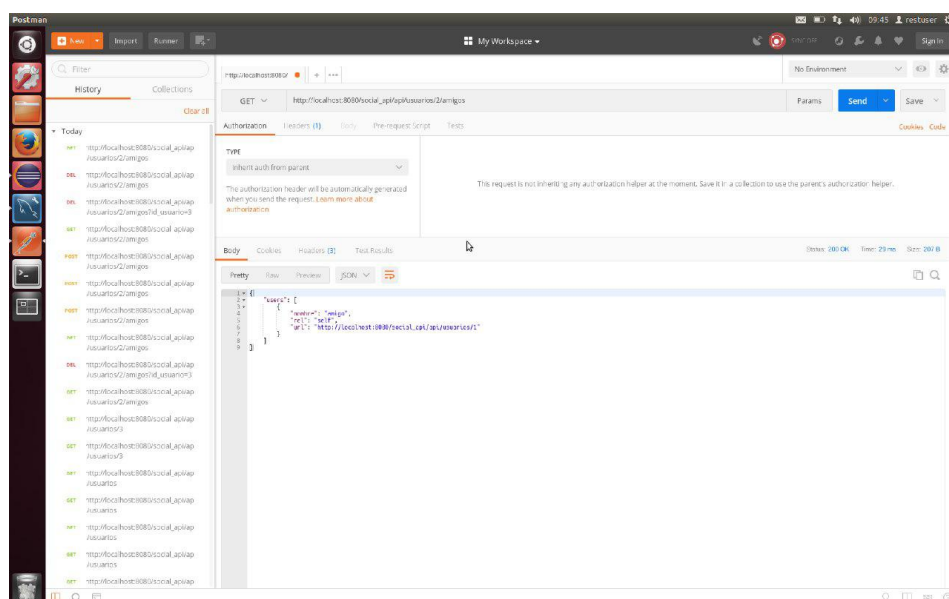


Figura 18: Método GET para obtener la lista de amigos del usuario con id 2.

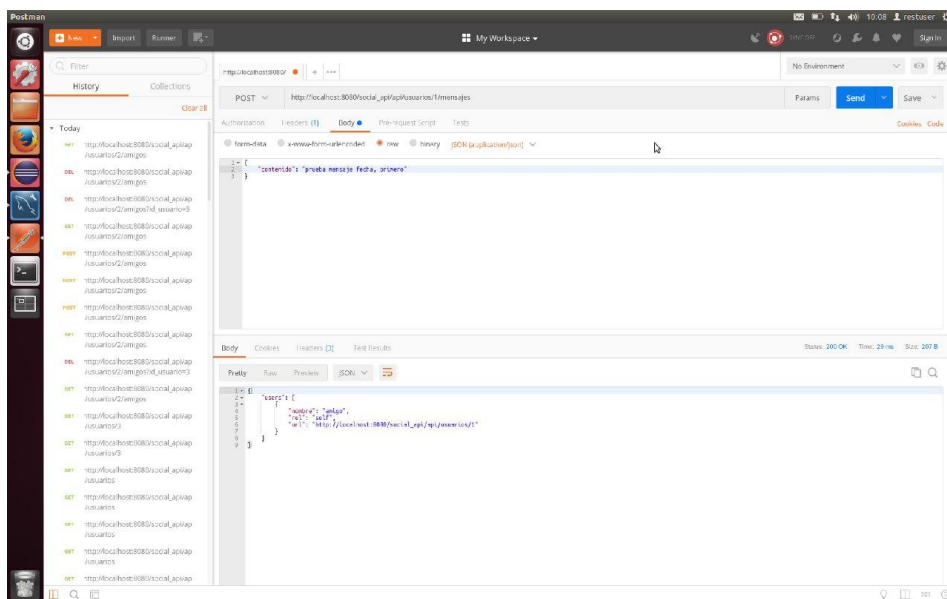


Figura 19: Método POST para añadir un mensaje en la página del usuario con id 1.

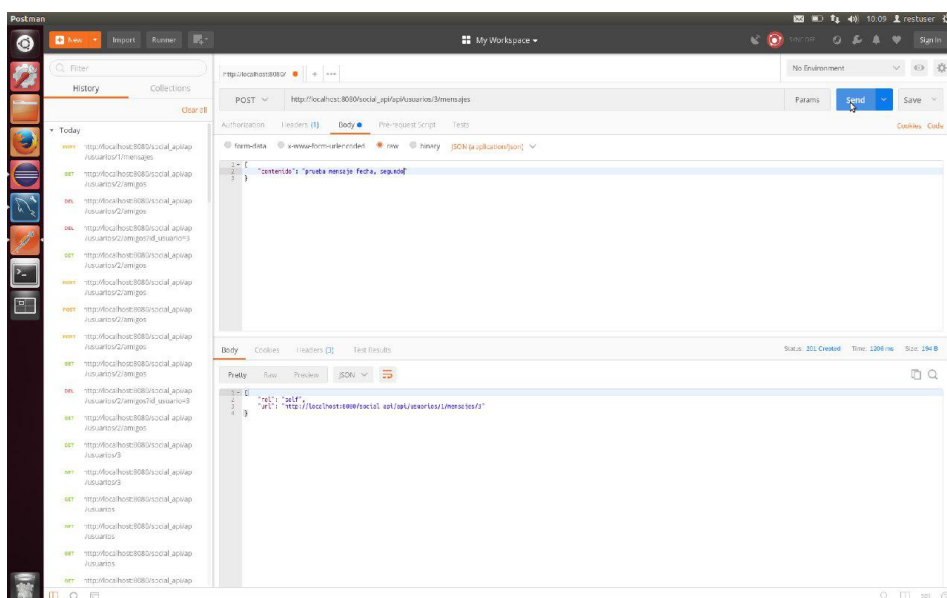


Figura 20: Método POST para añadir un mensaje en la página del usuario con id 3.

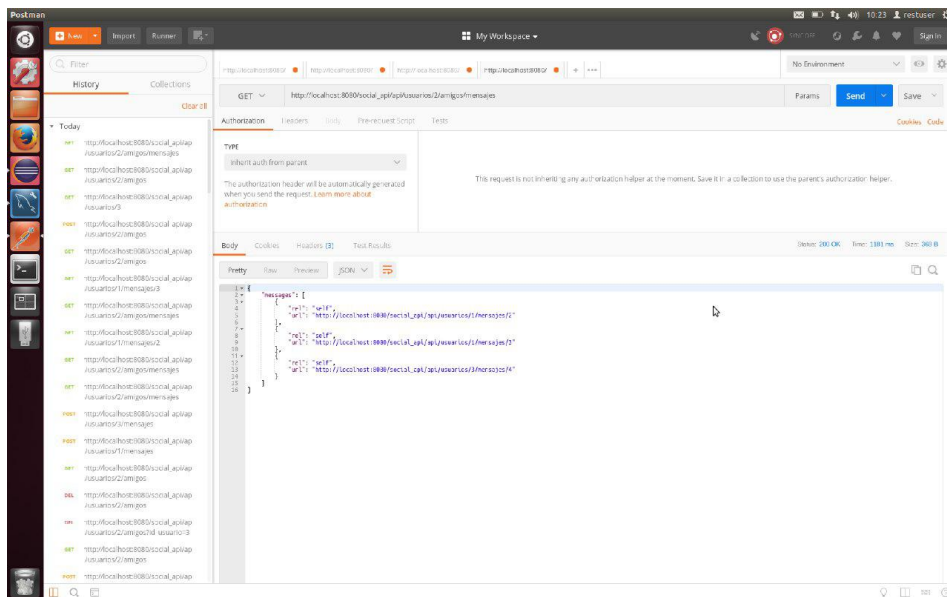


Figura 21: Método GET para obtener los mensajes de amigos del usuario con id 2.

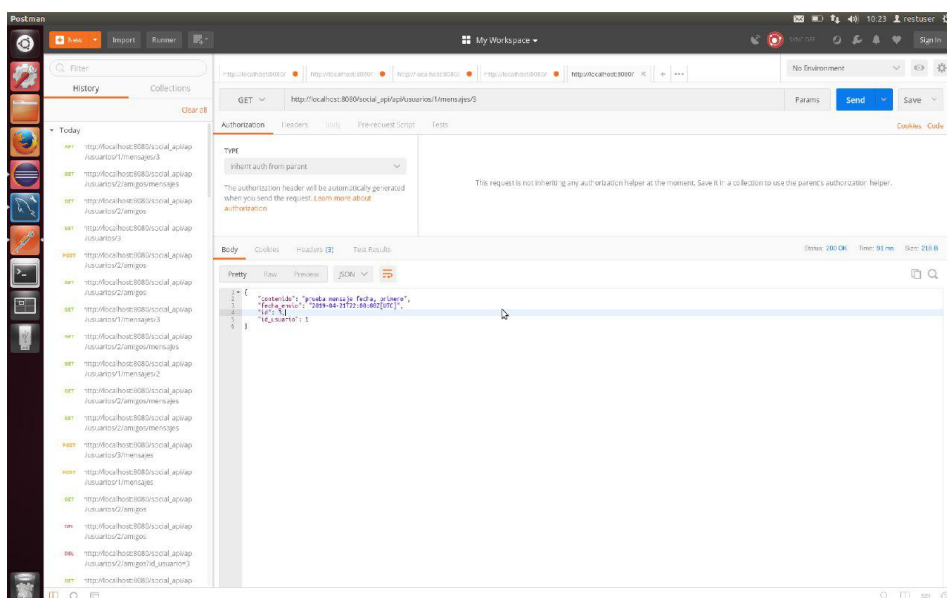


Figura 22: Método GET para obtener el mensaje con id 3 del usuario con id 1.

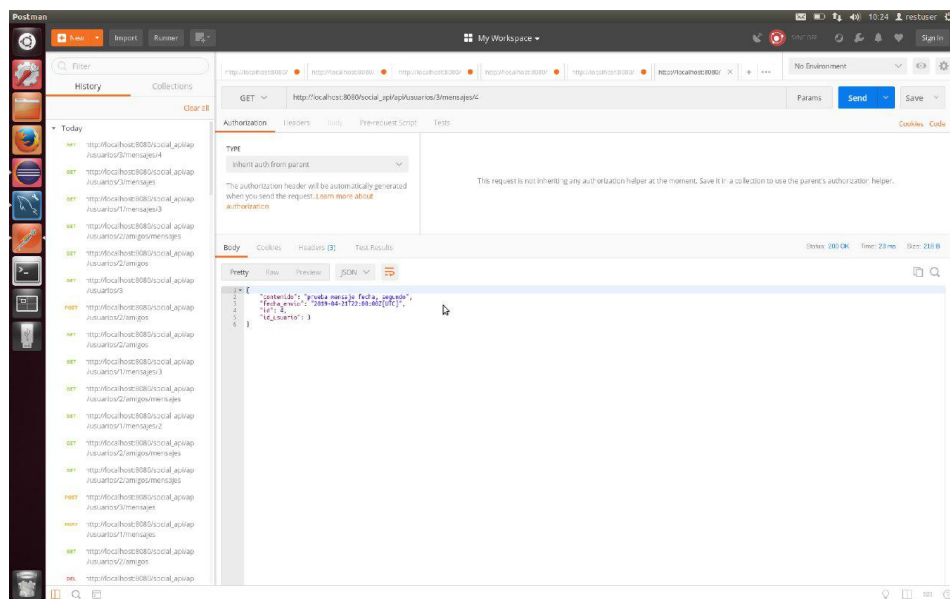


Figura 23: Método GET para obtener el mensaje con id 4 del usuario con id 3.

3. ANEXO: Problemas encontrados

Los fallos encontrados referentes a la Máquina virtual son:

- Con la aplicación Postman, al probar los métodos HTTP.
- Con la aplicación Eclipse, que algunas veces no encontraba las URIs implementadas.
- Con la baja calidad de las capturas de pantalla del SO utilizado (probablemente por la antigüedad del mismo).

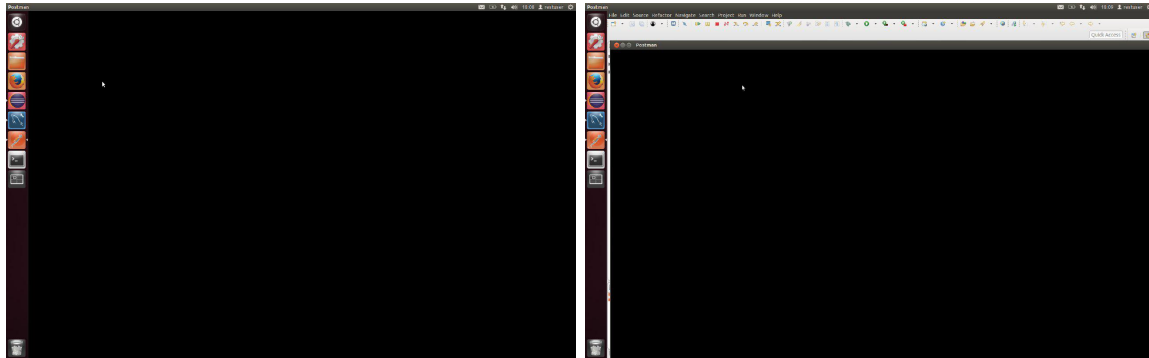


Figura 24: Fallo de la aplicación Postman al probar métodos.