# Trabajo Final:

# COMPUTACIÓN ORIENTADA A SERVICIOS


**Profesor:**

**Dr. Mauricio Arroqui**

**Integrantes:**

**Lacoste, Yamil (email: yamil.lacost@gmail.com).**

**San Roman, Matias (email: matias.sanroman.32@gmail.com).**

# Introducción:

Se crea un Web Services para el registro de usuarios y reciclados. La App tiene la siguiente funcionalidad:

- Registrar un usuario.
- Agregar un reciclaje a un usuario.
- Obtener todos los reciclajes de un usuario (sumados).
- Obtener el total reciclado.
- Listar todos los usuarios.
- Según un ID te da la información del usuario.
- Según un USERNAME te da la información del usuario.
- Listar todos los reciclados.
- Dado un ID de usuario me da todos los reciclados que hizo ese usuario.

# Recursos:

Se utilizo el framework de Sprint Boot, MySQL Workbench como bases de datos, Hibernate (mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación) y Postman para testear la App.

Aclaraciones: como no se usó una base de datos en memoria, se encuentra un archivo llamado Scripts, donde están las estructuras de Usuario y Reciclado y algunas inserciones. Tanto Usuario como Reciclado tienen un auto incremental en su ID.

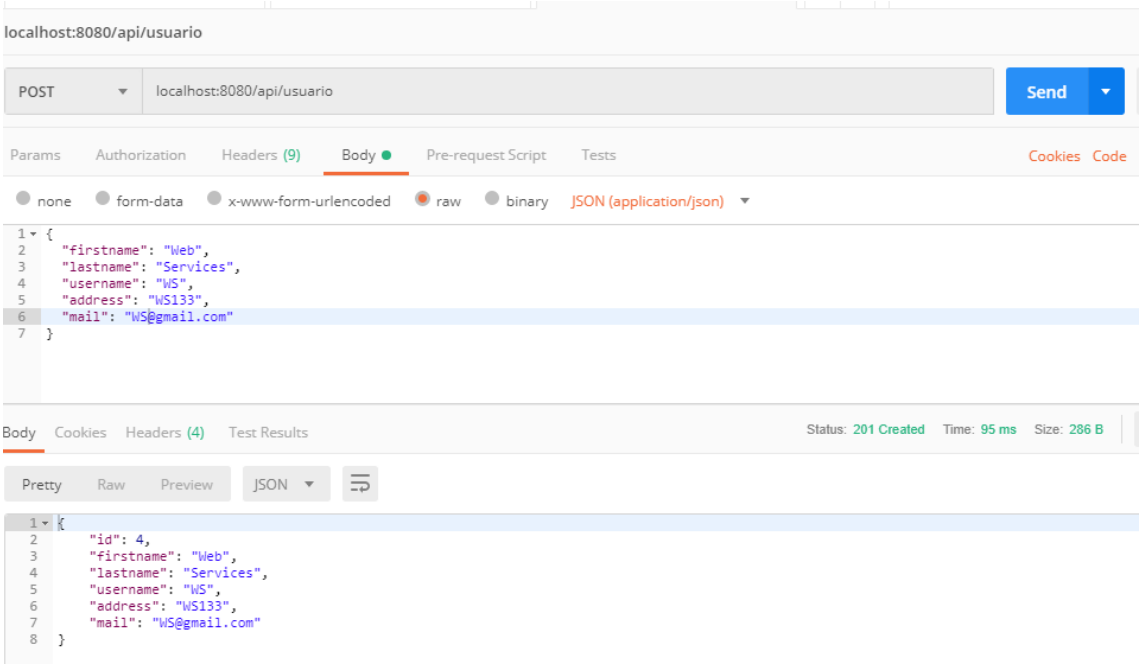
# Desarrollo:

Se creo el proyecto en Sprint Boot con las dependencias Web, Web tools, Mysql, JPA y Rest Repositories.

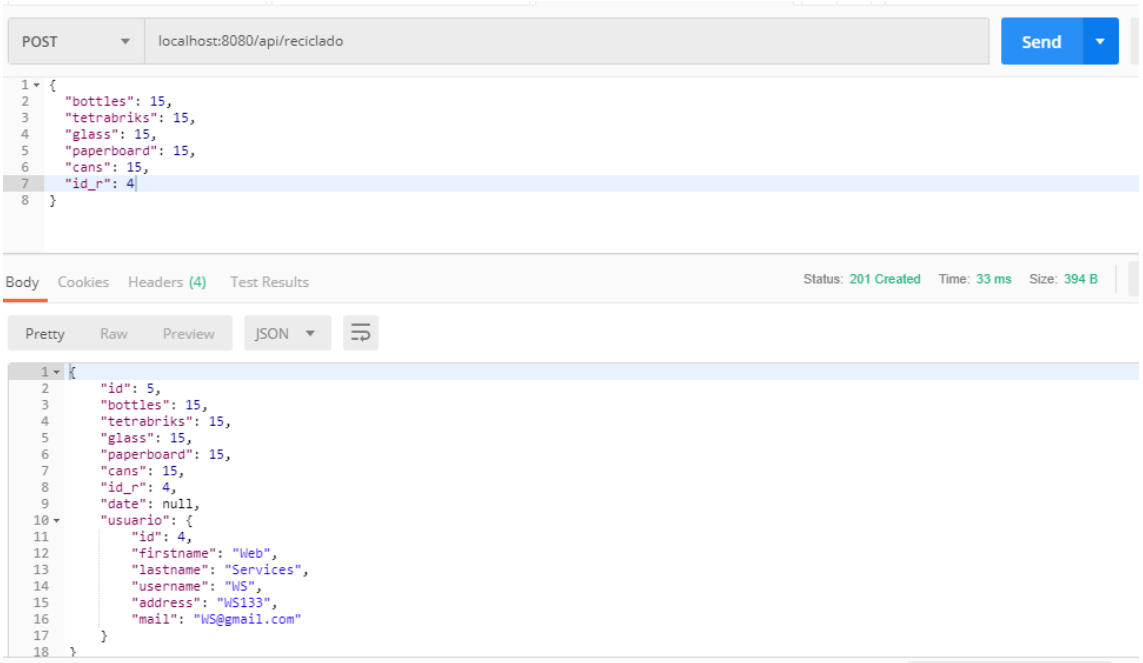
En modelos se crearon 4 clases, una para Usuario con todos los datos, UsuarioReciclado donde se crea el reciclado y se le asocia un Usuario para esto se utilizó la anotación @OneToOne y @JoinColumn donde se indica la cardinalidad y porque columnas haces la unión. TotalMaterial que se va a utilizar para devolver todos los reciclajes de un usuario y su suma. Y por último TotalReciclaje que se usa para devolver la suma de cada material de todos los usuarios y, además, se declaró una constante Factor que calcula las toneladas.
Se crea un paquete de Repositorios con dos clases Usuario y UsuarioReciclado que extienden de JpaRepository para hacerlos persistentes en la Base de Datos, se utilizan dos métodos, uno para buscar por ID y otro método para devolver todos.
El siguiente paquete es de Servicios donde se implementa la lógica del negocio con sus respectivos métodos.
Por último, el paquete de Controlador donde se juntan las peticiones HTTP y los servicios.

A continuación, se muestran fotos de cada método con su mensaje y sus respuestas a través de Postman. Después de insertar Usuario y Reciclado se muestran los datos de la Base de Datos en sus respectivas imágenes.

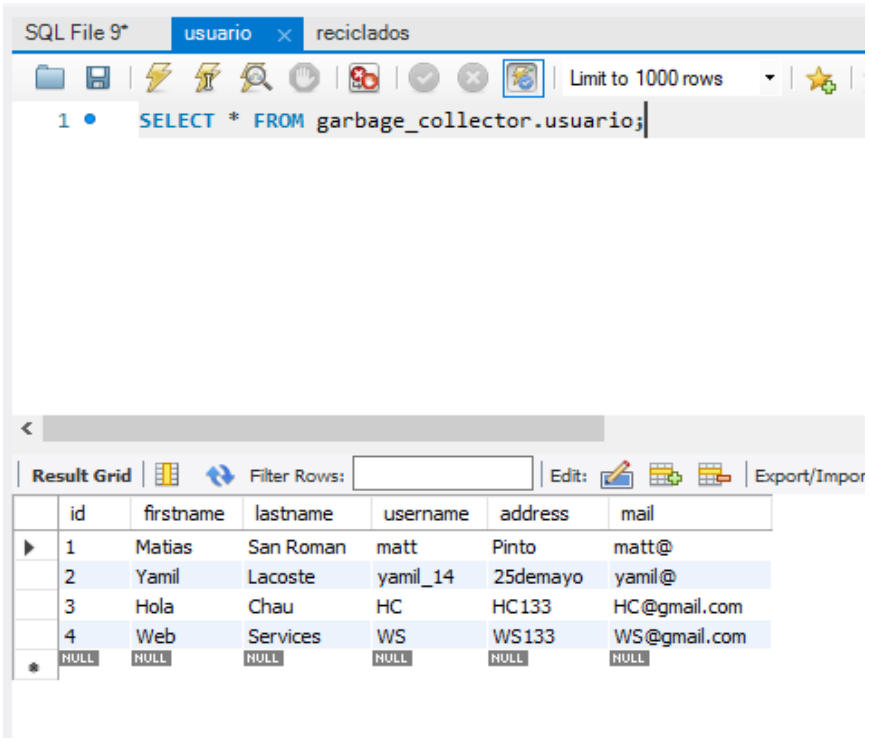
## Registrar un usuario (@PostMapping(path = "/api/usuario")).



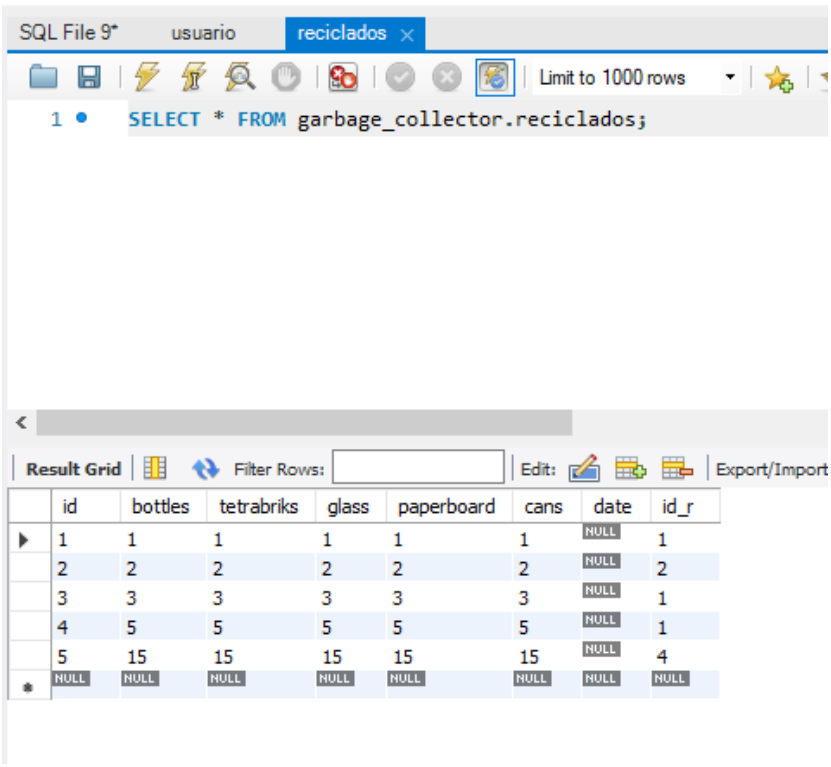

## Agregar un reciclaje a un usuario (@PostMapping(path = "/api/reciclado")).



## Datos de la Base de Datos:

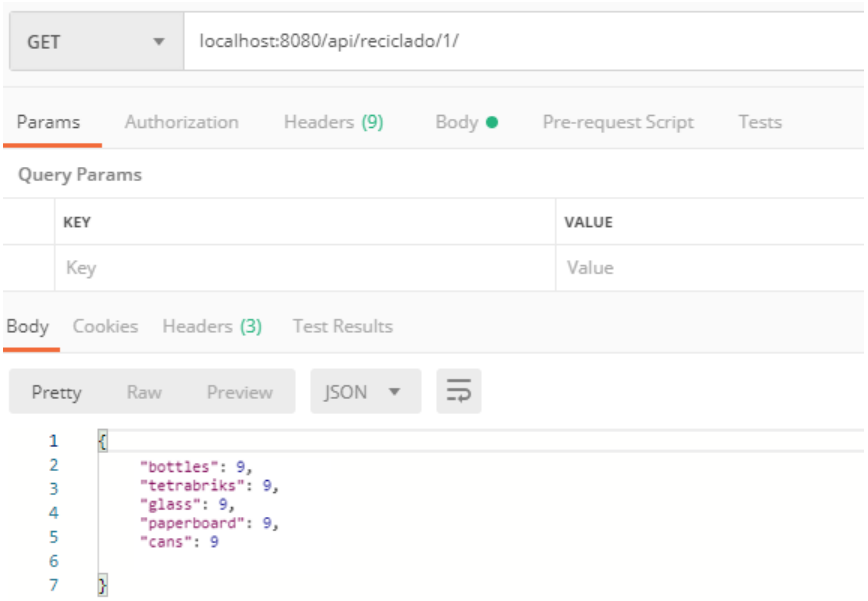



| id | firstname | lastname | username | address | mail |
|---|---|---|---|---|---|
| 1 | Matias | San Roman | matt | Pinto | matt@ |
| 2 | Yamil | Lacoste | yamil_14 | 25demayo | yamil@ |
| 3 | Hola | Chau | HC | HC133 | HC@gmail.com |
| 4 | Web | Services | WS | WS133 | WS@gmail.com |
| NULL | NULL | NULL | NULL | NULL | NULL |

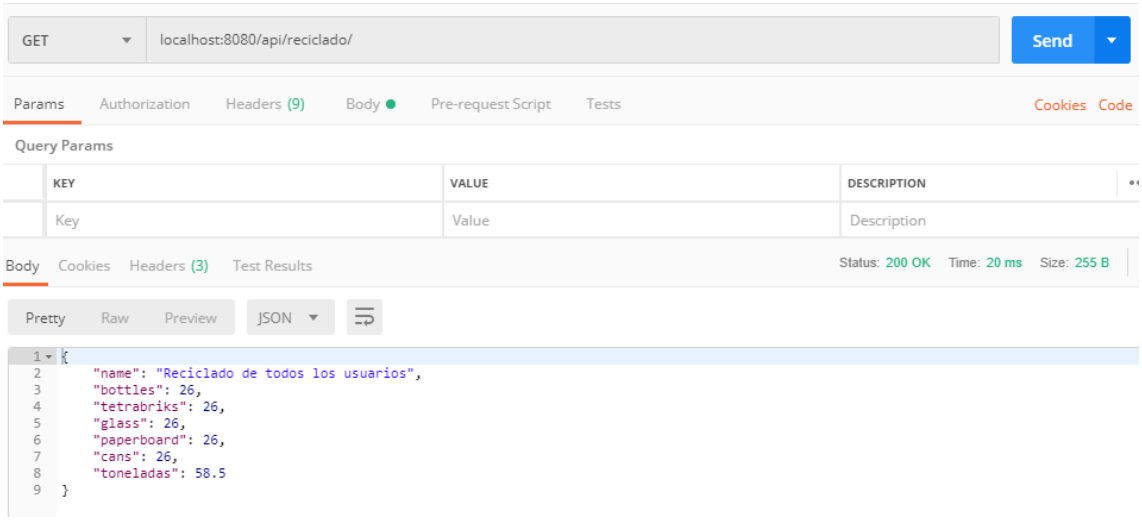



| id | bottles | tetrabriks | glass | paperboard | cans | date | id_r |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | NULL | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 | NULL | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 | NULL | 1 |
| 4 | 5 | 5 | 5 | 5 | 5 | NULL | 1 |
| 5 | 15 | 15 | 15 | 15 | 15 | NULL | 4 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

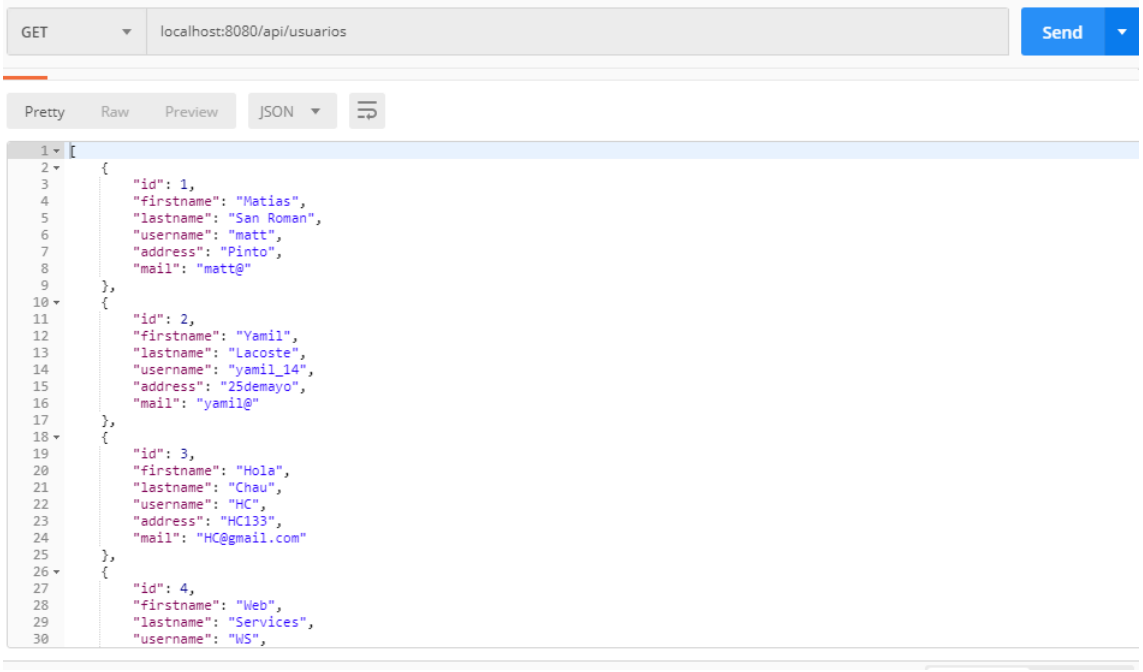
**Obtener todos los reciclajes de un usuario (sumados) (@GetMapping(path = "/api/reciclado/{name}/")).**



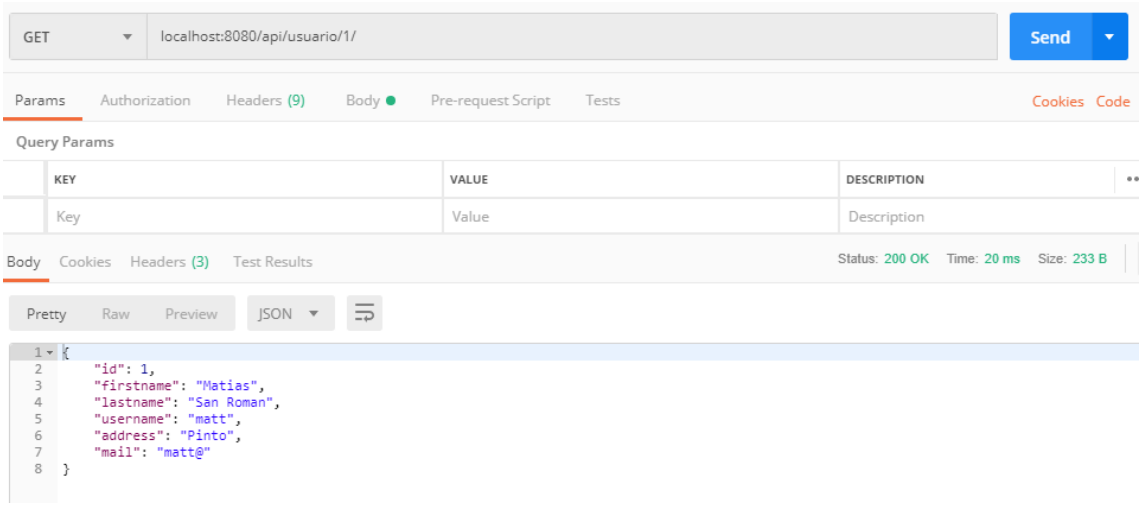

**Obtener el total reciclado (@GetMapping(path = "/api/reciclado/")).**



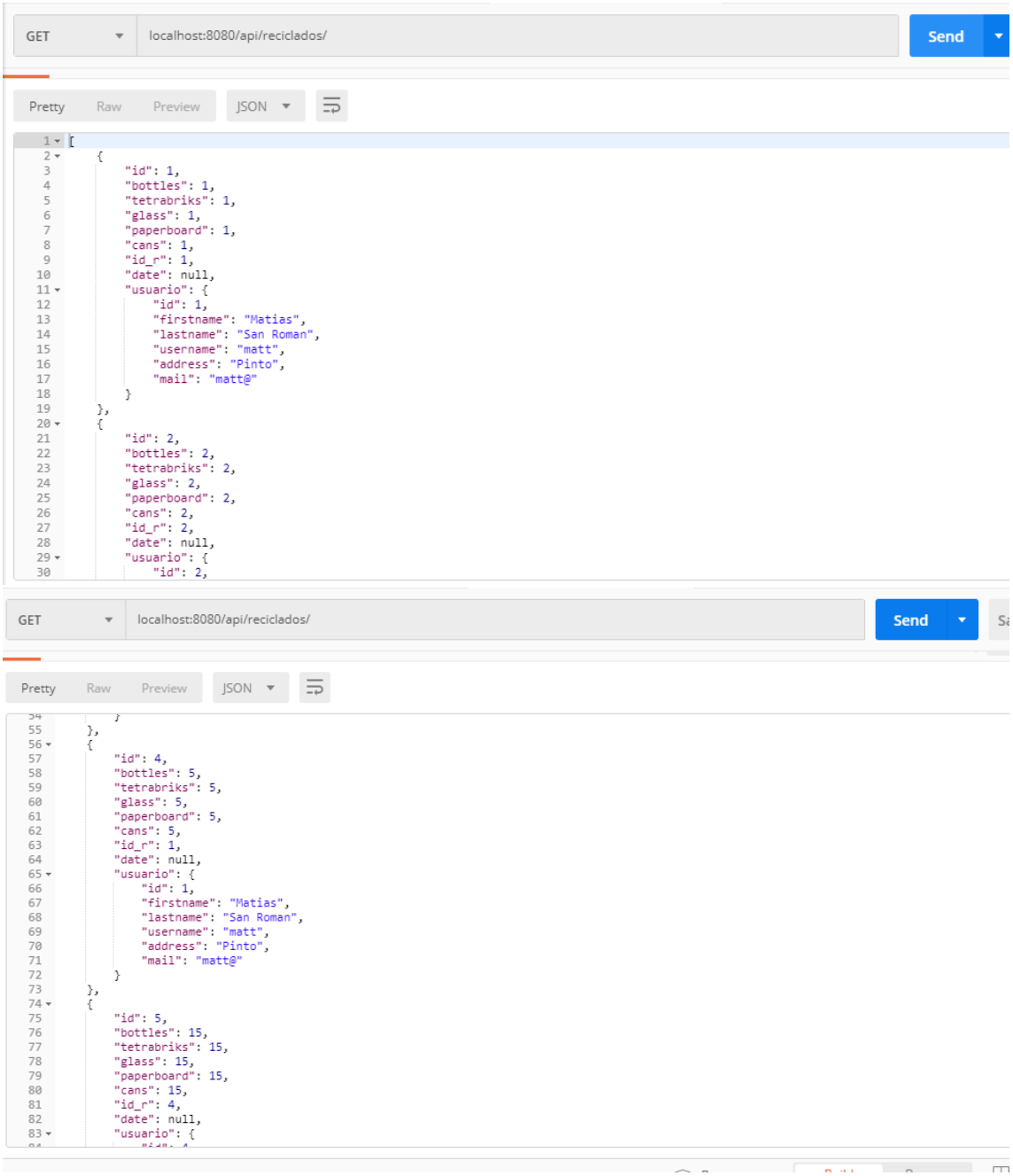
**Listar todos los usuarios (@GetMapping(path = "/api/usuarios")).**



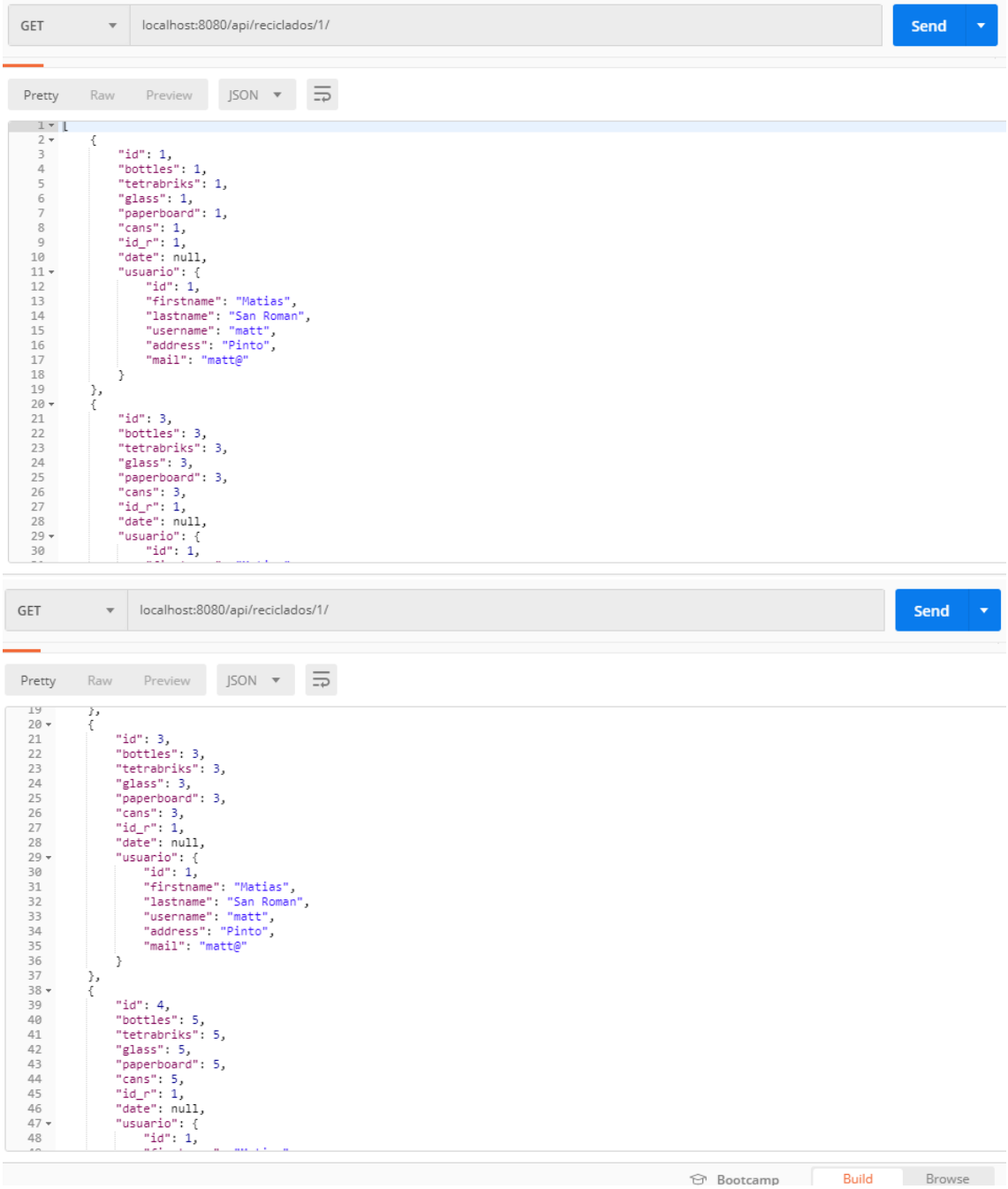

**Según un ID te da la información del usuario (@GetMapping(path = "/api/usuario/{name}/")).**



## Listar todos los reciclados (@GetMapping(path = "/api/reciclados")).



**Dado un ID de usuario me da todos los reciclados que hizo ese usuario (@GetMapping(path = "/api/reciclados/{name}/")).**



# Conclusión:

Se resolvió el enunciado propuesto por la cursada, donde tuvimos que aprender sobre objetos JSON y la facilidad de estos para intercambiar información. Aprendimos a como relacionar los atributos de una clase java con los atributos de una base de datos gracias a HIBERNATE y como estos interactúan entre sí. Para poder hacer esta interacción se recurre al protocolo HTTP y sus métodos (GET, POST..).

Se ha testeado la APP desde PostMan, y como se pueden ver en las imágenes de arriba se pudo resolver correctamente el enunciado.