**Trabajo Final:**

**COMPUTACIÓN ORIENTADA A SERVICIOS**

****

**Integrantes:**

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

**San Roman, Matias (email:** [**matias.sanroman.32@gmail.com**](mailto: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.
* 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 y Postman para testear la App.

**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 ultimo TotalReciclaje que se usa para devolver la suma de cada material de todos los usuarios y además, se declaro una constante Factor que calcula las toneladas.

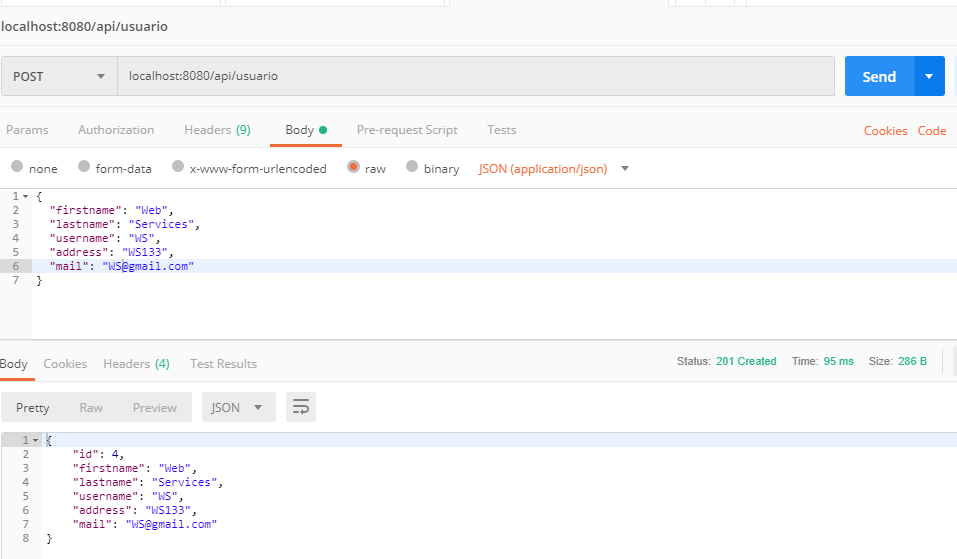
Se crea un paquete de Repositorios con dos clases Usuario y UsuarioReciclado que extienden de JpaRepository para hacerlos persistentes en la Base de Datos, se utilizan dos métodos, uno para buscar por ID y otro método para devolver todos.

El siguiente paquete es de Servicios donde se implementa la lógica del negocio con sus respectivos metodos.

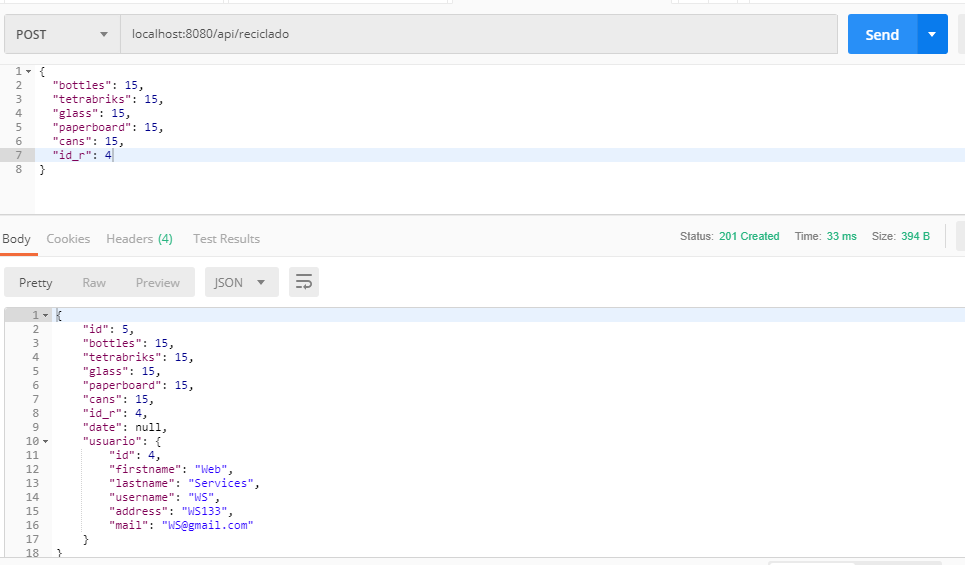
Por último, el paquete de Controlador donde se juntan las peticiones HTTP y los servicios.

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

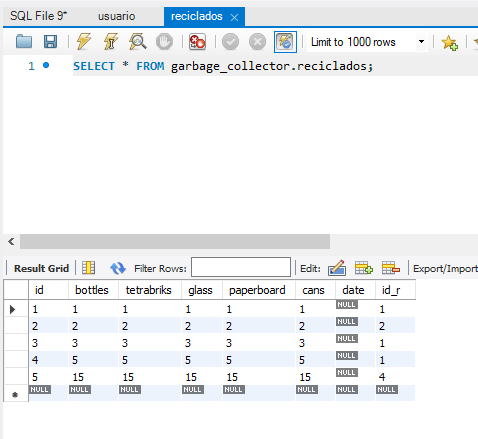
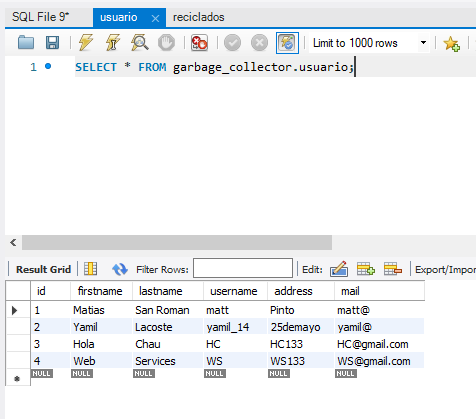
**Registrar un usuario (@PostMapping(path = "/api/usuario")).**

****

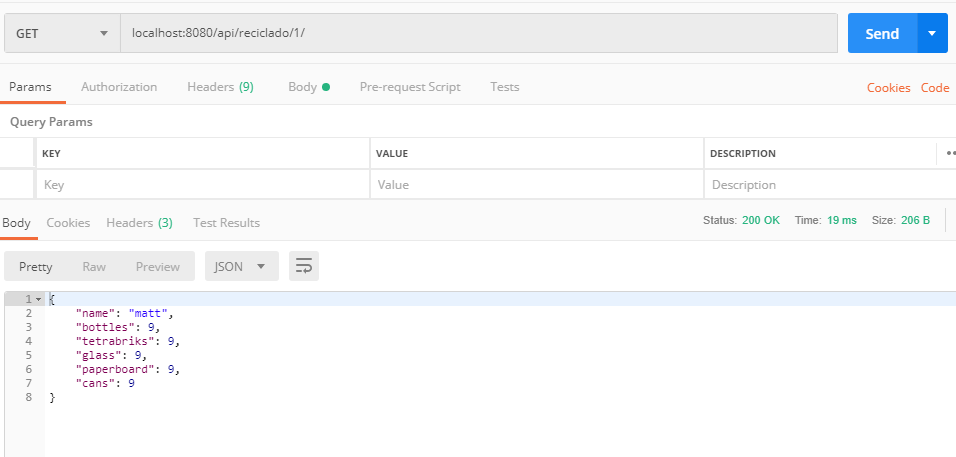
**Agregar un reciclaje a un usuario (@PostMapping(path = "/api/reciclado")).**

****

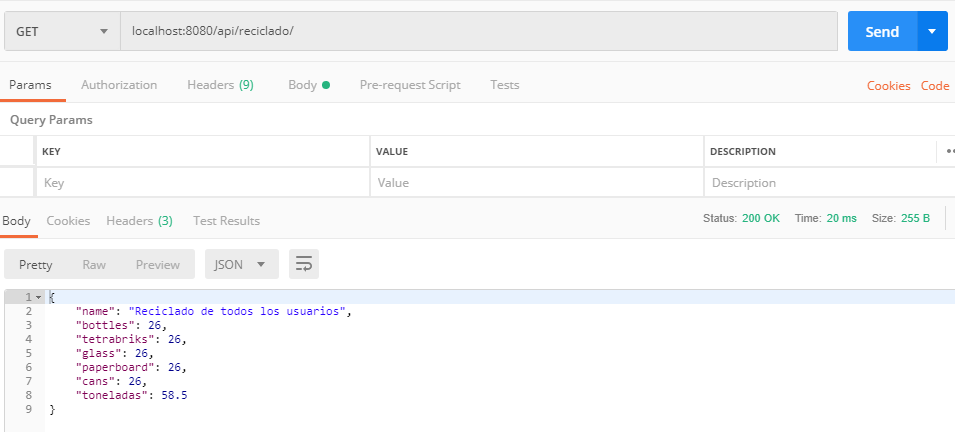
**Datos de la Base de Datos:**

****

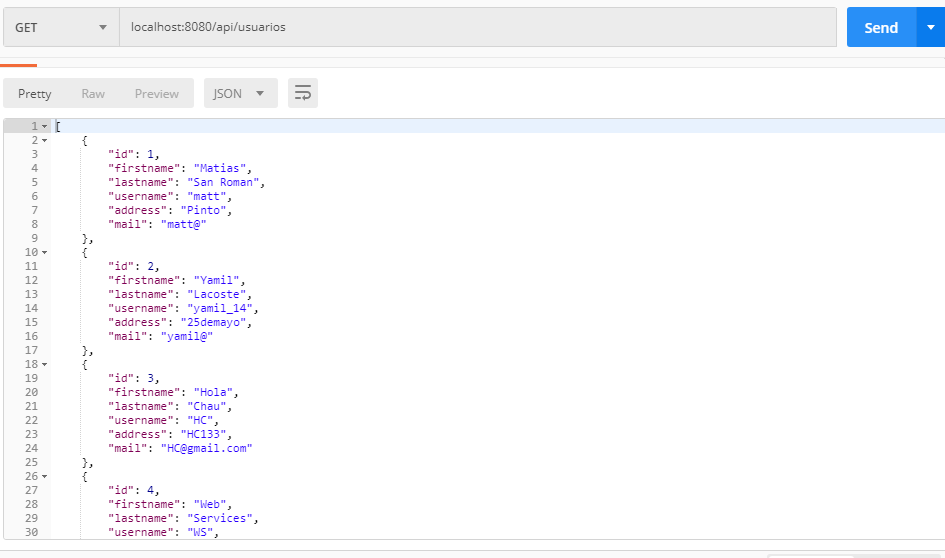
**Obtener todos los reciclajes de un usuario (sumados) (@GetMapping(path = "/api/reciclado/{name}/")).**

****

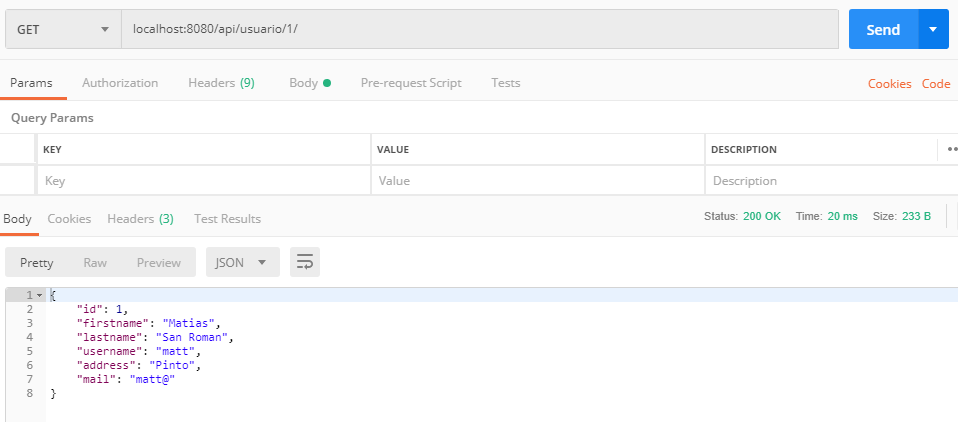
**Obtener el total reciclado (@GetMapping(path = "/api/reciclado/")).**

****

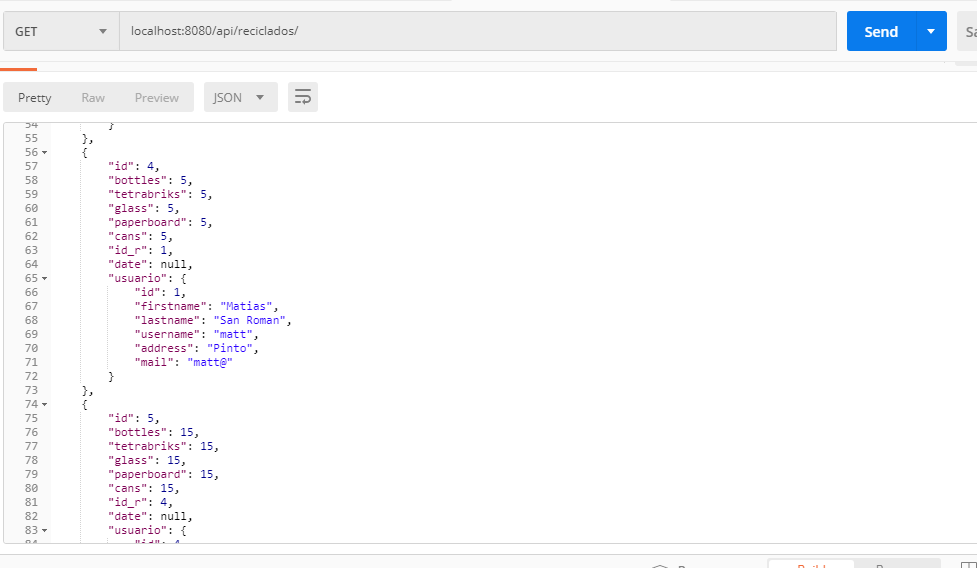
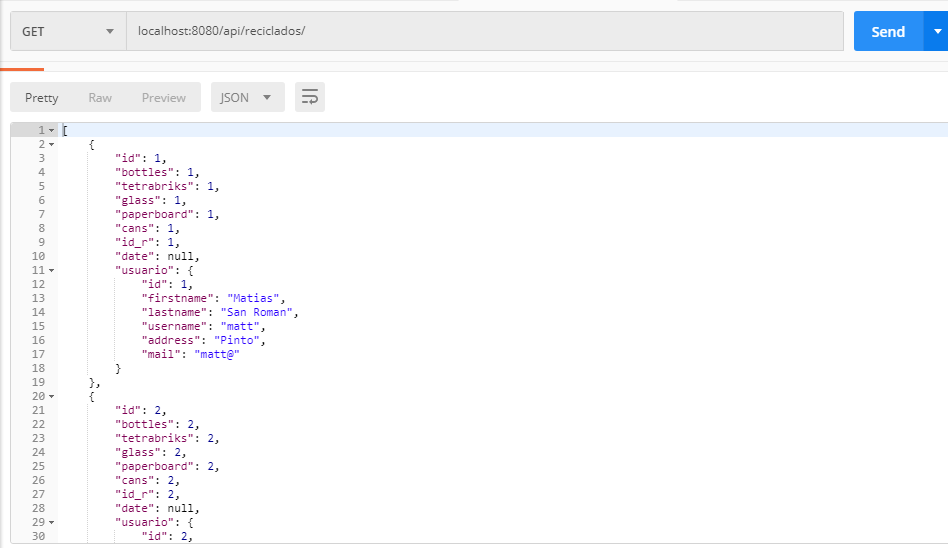
**Listar todos los usuarios (@GetMapping(path = "/api/usuarios")).**

****

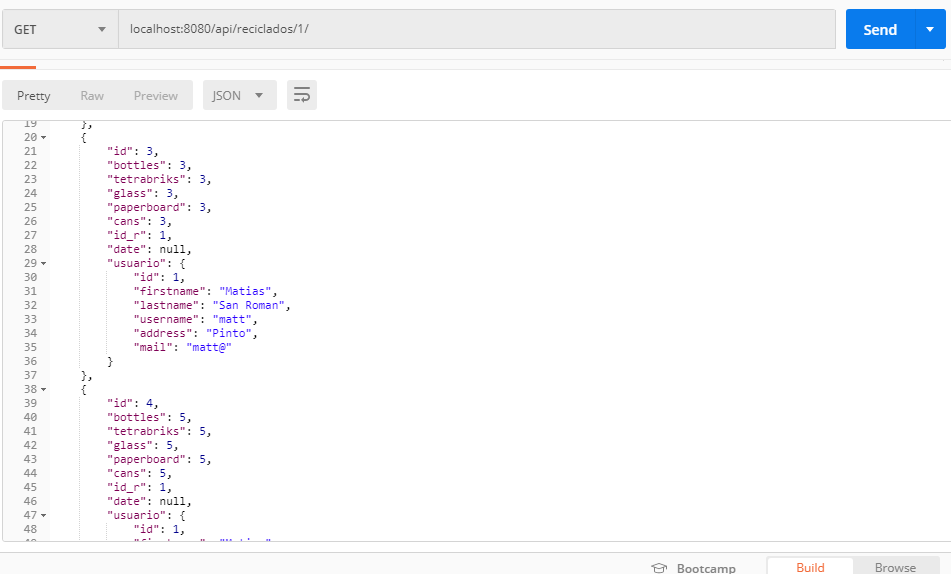
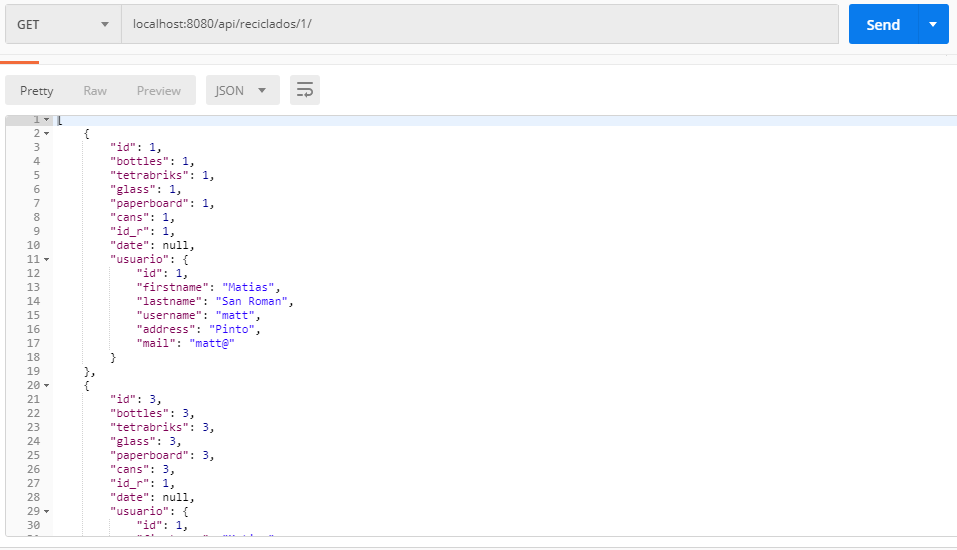
**Según un ID te da la información del usuario (@GetMapping(path = "/api/usuario/{name}/")).**

****

**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}/")).**

****