

Practica sps

PARte 1

Services & Processes Solutions

Tabla de contenido

[Eduardo Isaac Gonzalez Alvarez. 3](#_Toc31104036)

[Objetivo 3](#_Toc31104037)

[Endpoint. 5](#_Toc31104038)

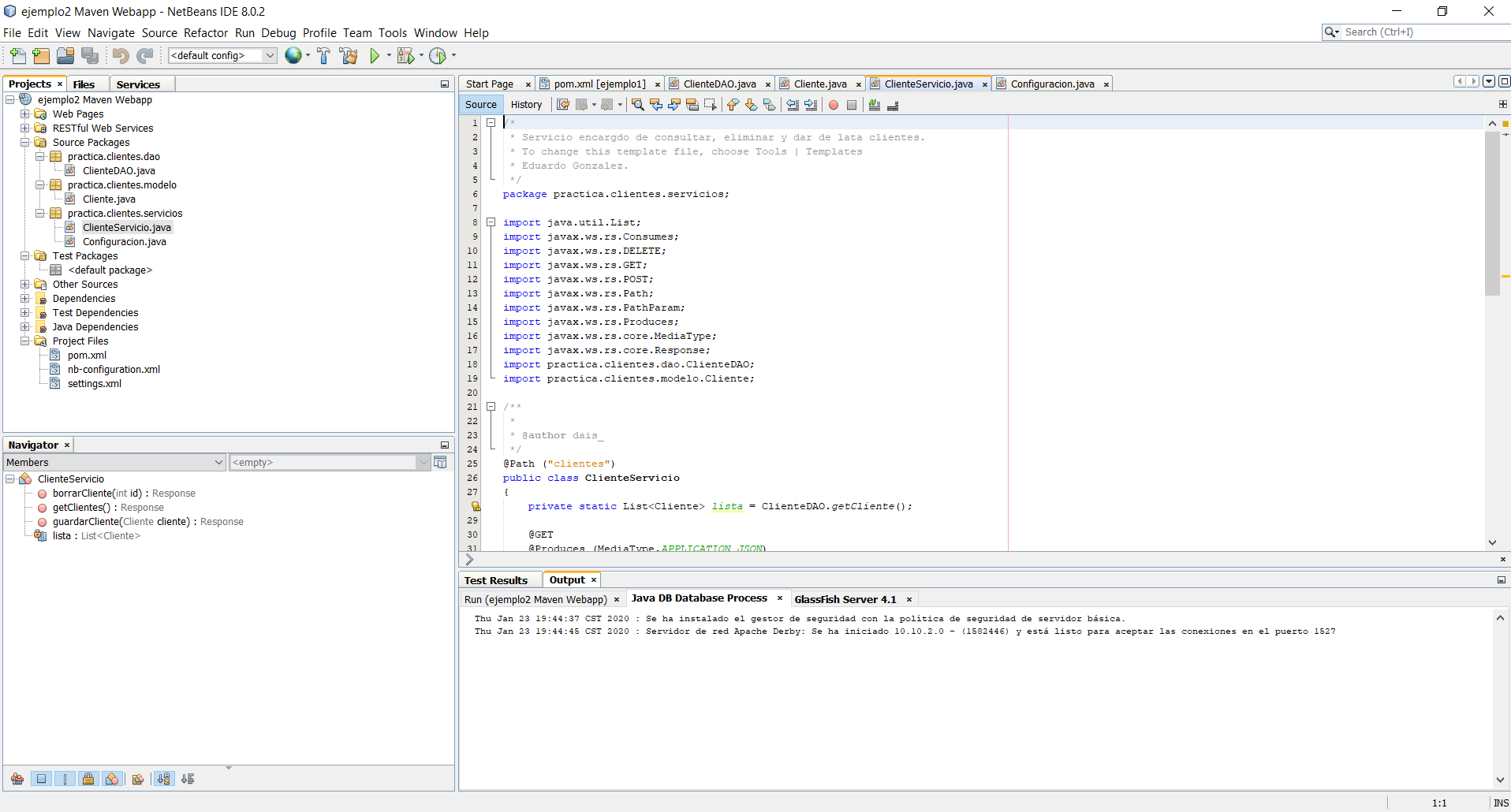
[Contenedor. 6](#_Toc31104039)

[GitHub 10](#_Toc31104040)

[Repositorio git. 11](#_Toc31104041)

# Eduardo Isaac Gonzalez Alvarez.

El lenguaje de programación que se escogió para el desarrollo del servicio fue java con NETBEANS 8.0.2, el formato es Json.



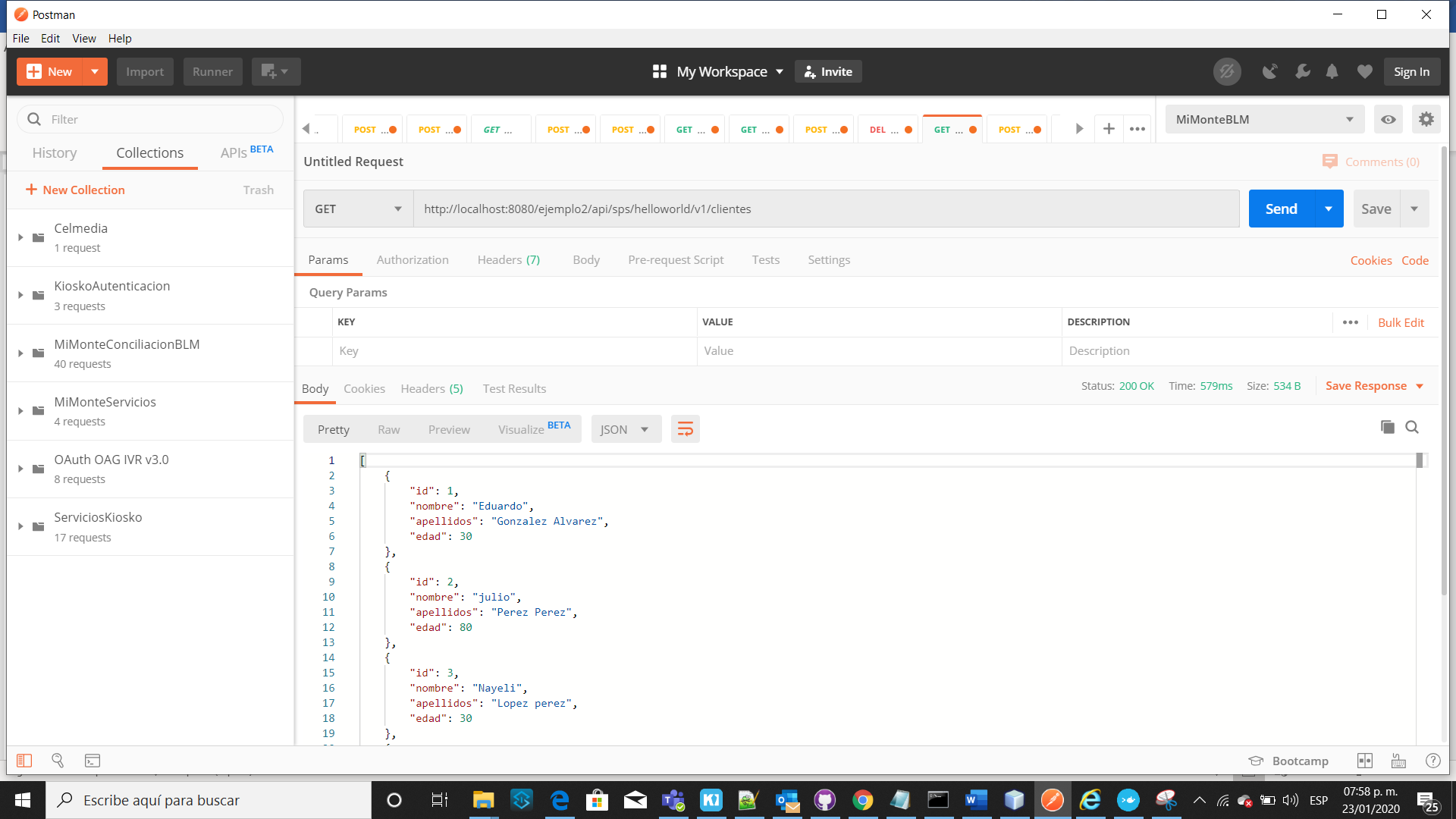
# Objetivo

El servicio se encarga de la consulta, eliminación y creación de Clientes.

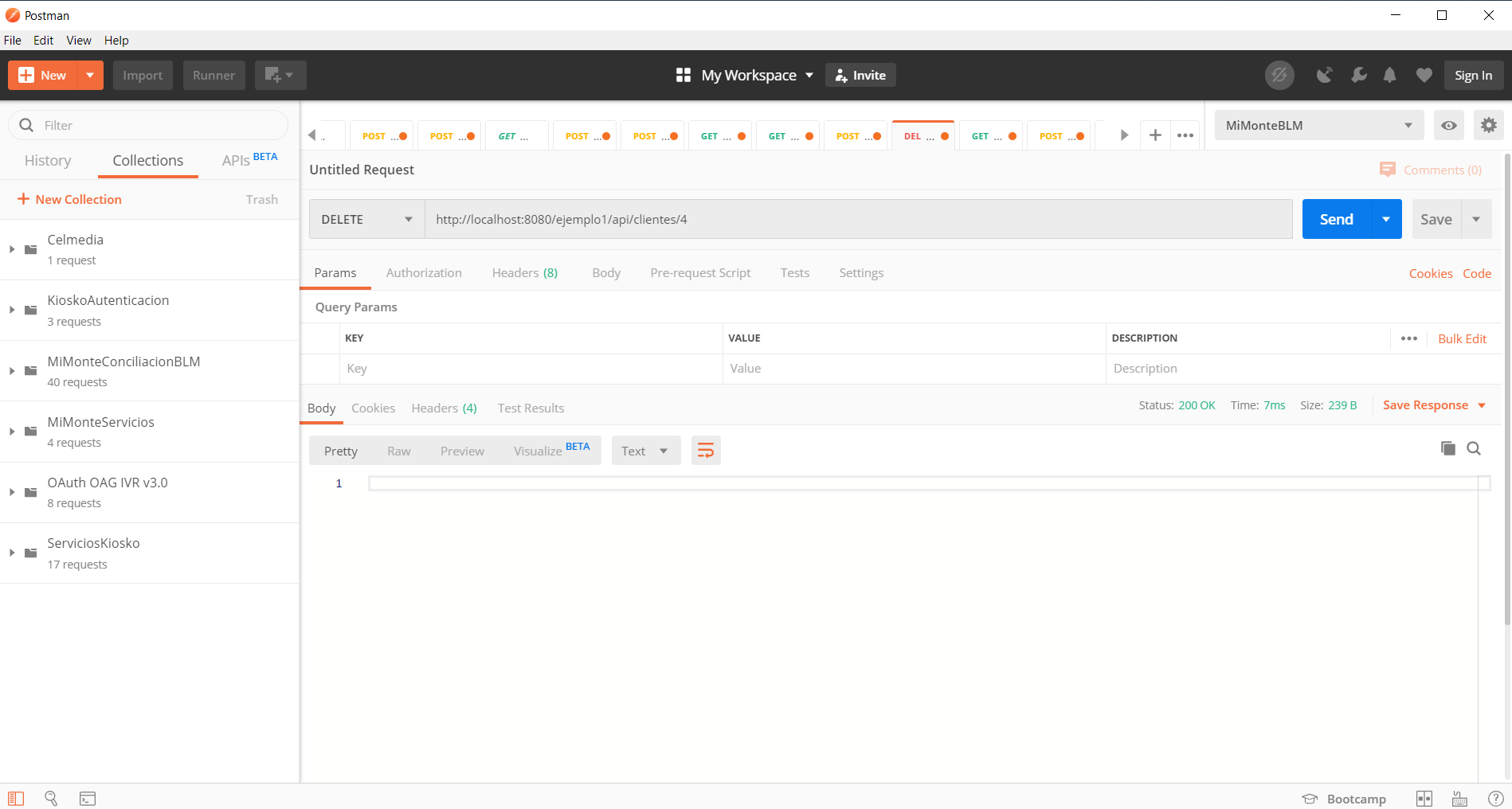
Pruebas Postman

Consulta de clientes.

También se debe de exponer dicha aplicación como un micro servicio.

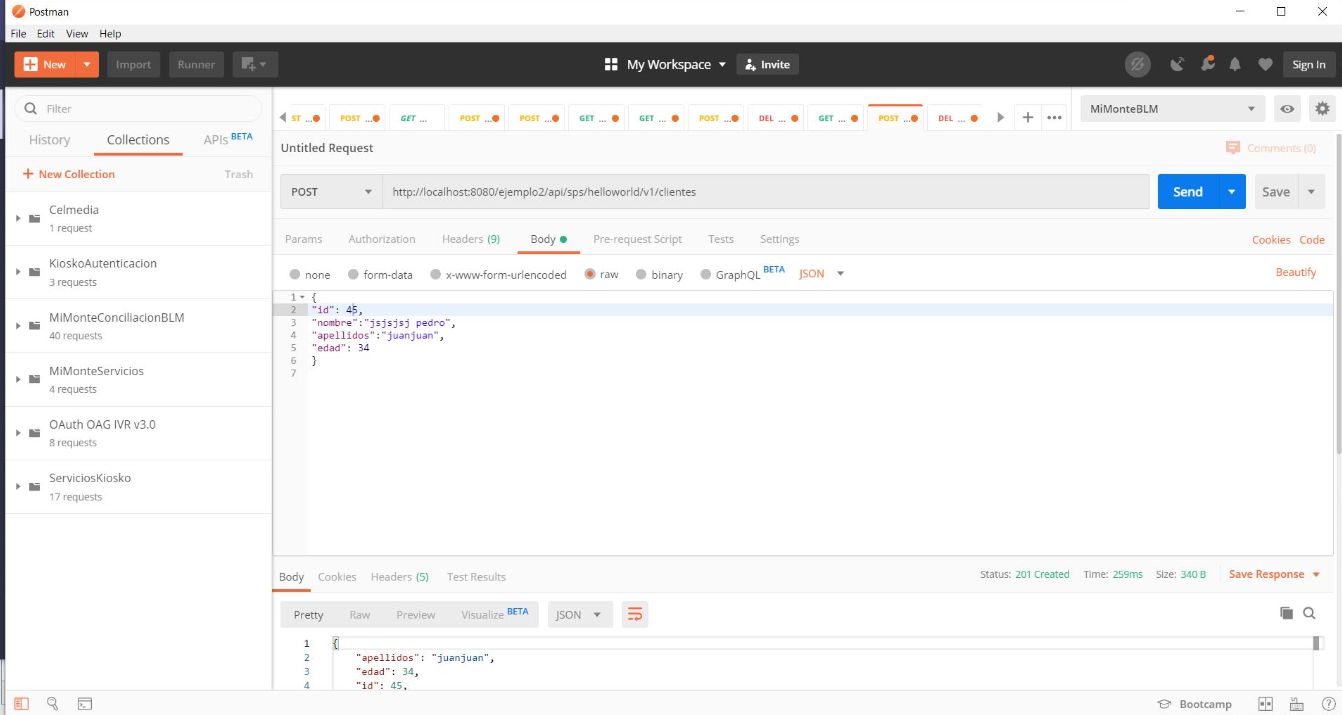


Eliminacion de Clientes.



Servicios POST

Creación de Clientes.



# Endpoint.

Get

<http://localhost:8080/ejemplo2/api/sps/helloworld/v1/clientes>

Post

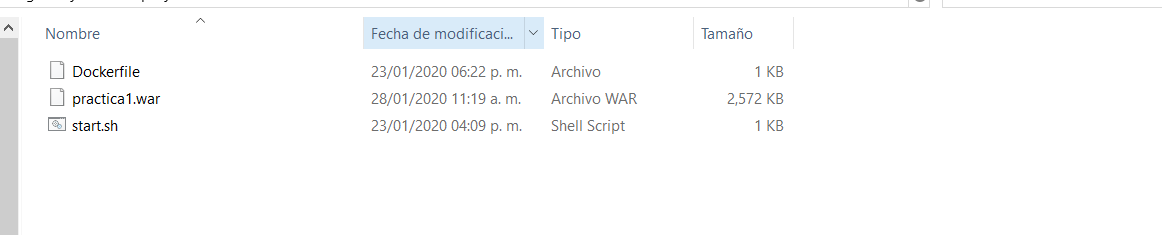
<http://localhost:8080/ejemplo2/api/sps/helloworld/v1/clientes>

Delete

<http://localhost:8080/ejemplo2/api/clientes/55>

# Contenedor.

1.- Se crea una carpeta con los archivos .war, .sh y Doker file

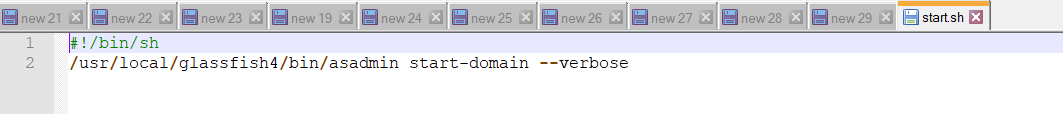


Anterior mente el Docker file lo tenia configurado de la sig manera.

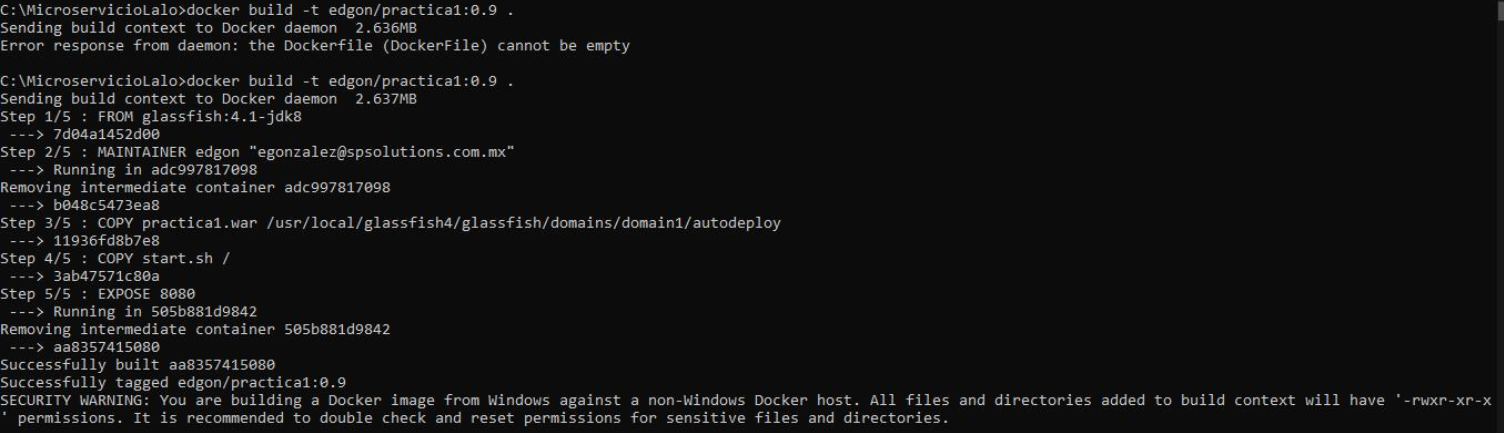
FROM glassfish:4.1-JDK8  
MAINTAINER edgon "egonzalez@spsolutions.com.mx"  
COPY practica1.war /  
COPY start.sh /  
EXPOSE 8080  
ENTRYPOINT ["/start.sh"]

De esta forma no me fue posible levanta el contenedor por lo que tome otro ejemplo, en el cual copiamos el archivo.war a la carpeta de autodeploy de glassfish4





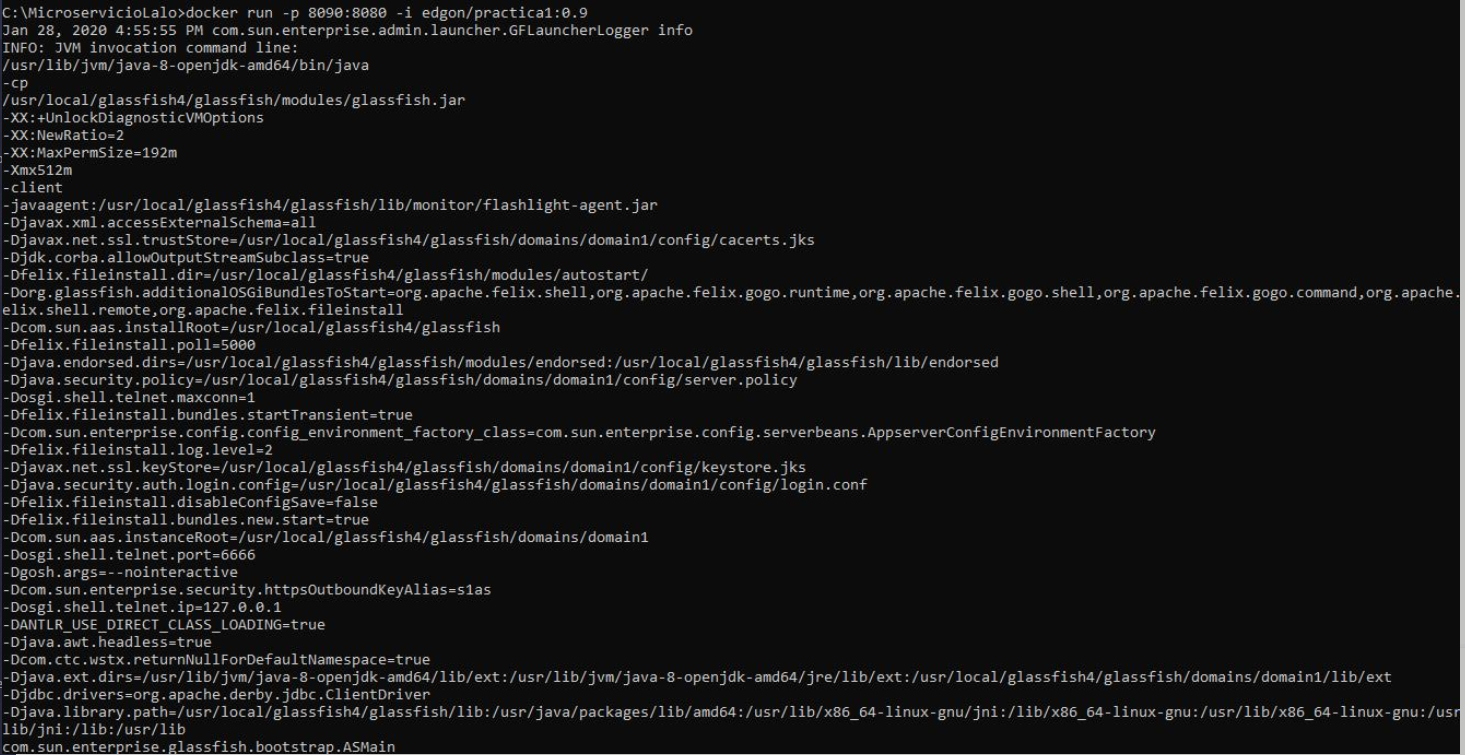
2.- Ahora procedemos a crear la imagen del contenedor.



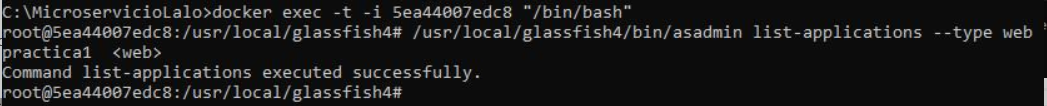
3.- verificamos si ya se encuentra en nuestras imágenes.



4.- Procedemos a desplegar la aplicación y exponerla por el puerto 8090.

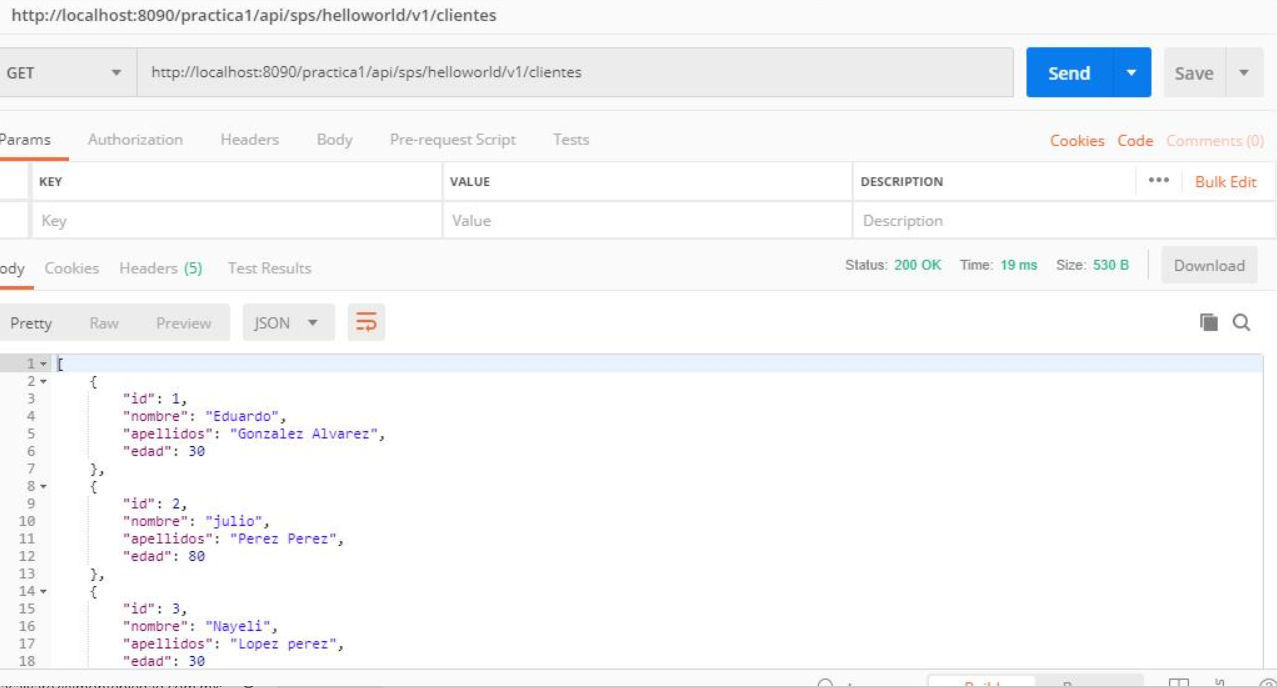


5.- con los siguientes 2 comandos entramos al contenedor y validamos que aplicacions web se encuentran trabajando.

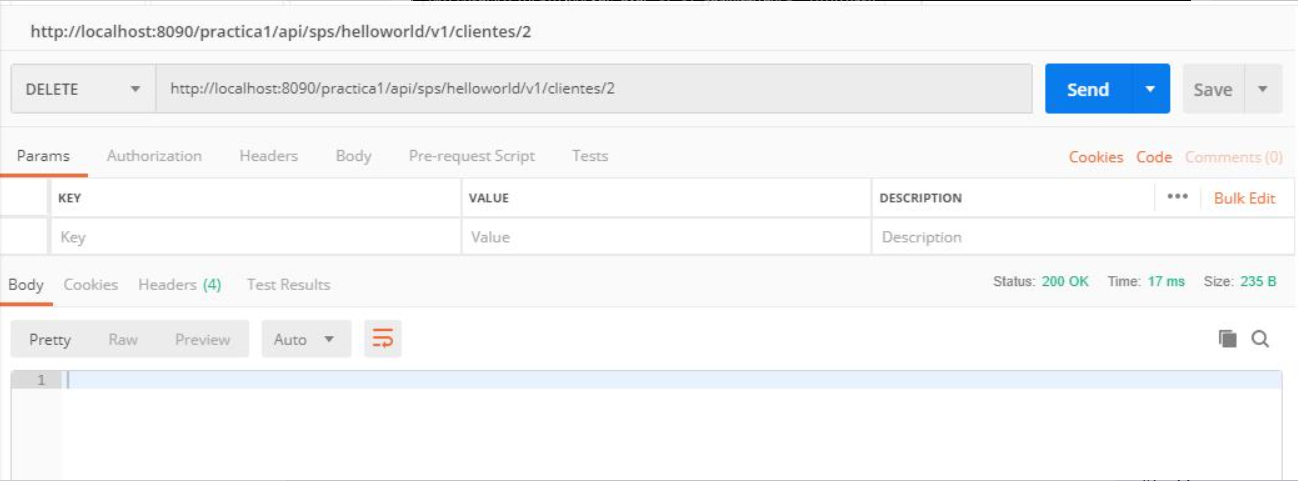


6.- Ahora podemos realizar las pruebas de nuestro icroservicio, yo las realizo en postman

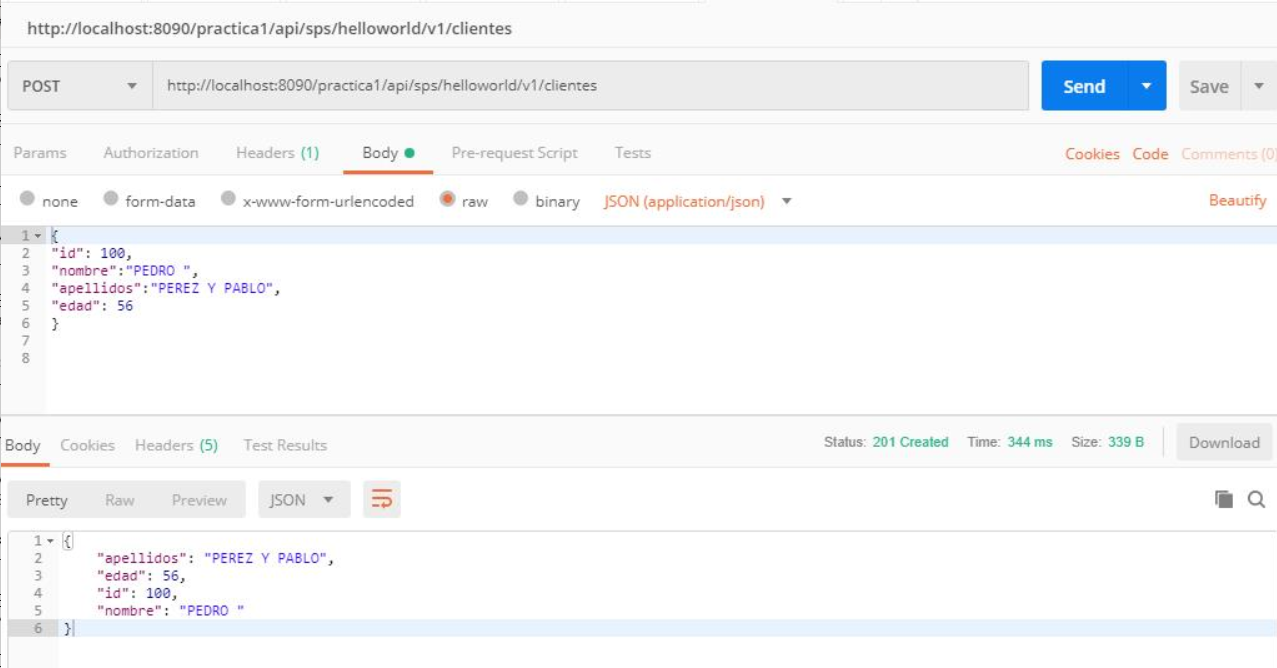
Busqueda de clientes.GET



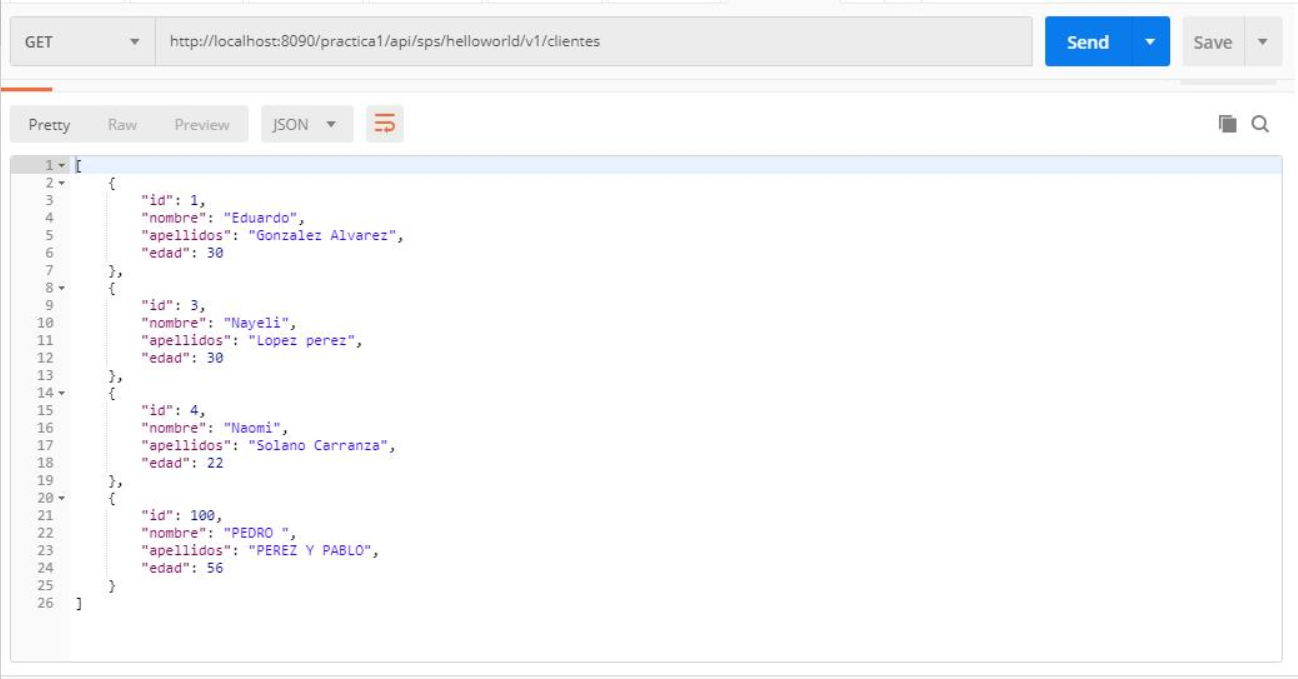
Eliminacion de cliente. DELETE



Creacion de un cliente .POST

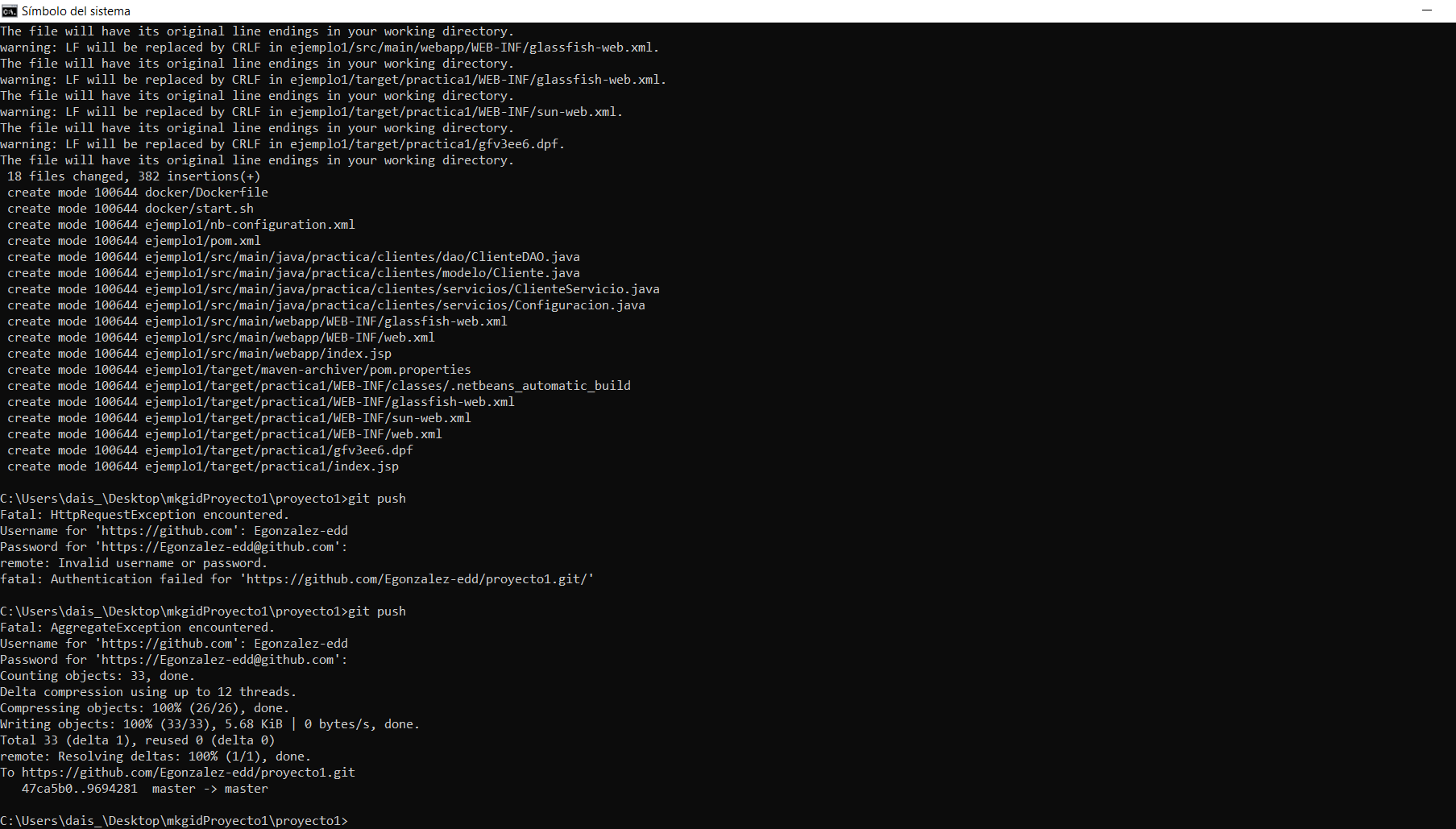


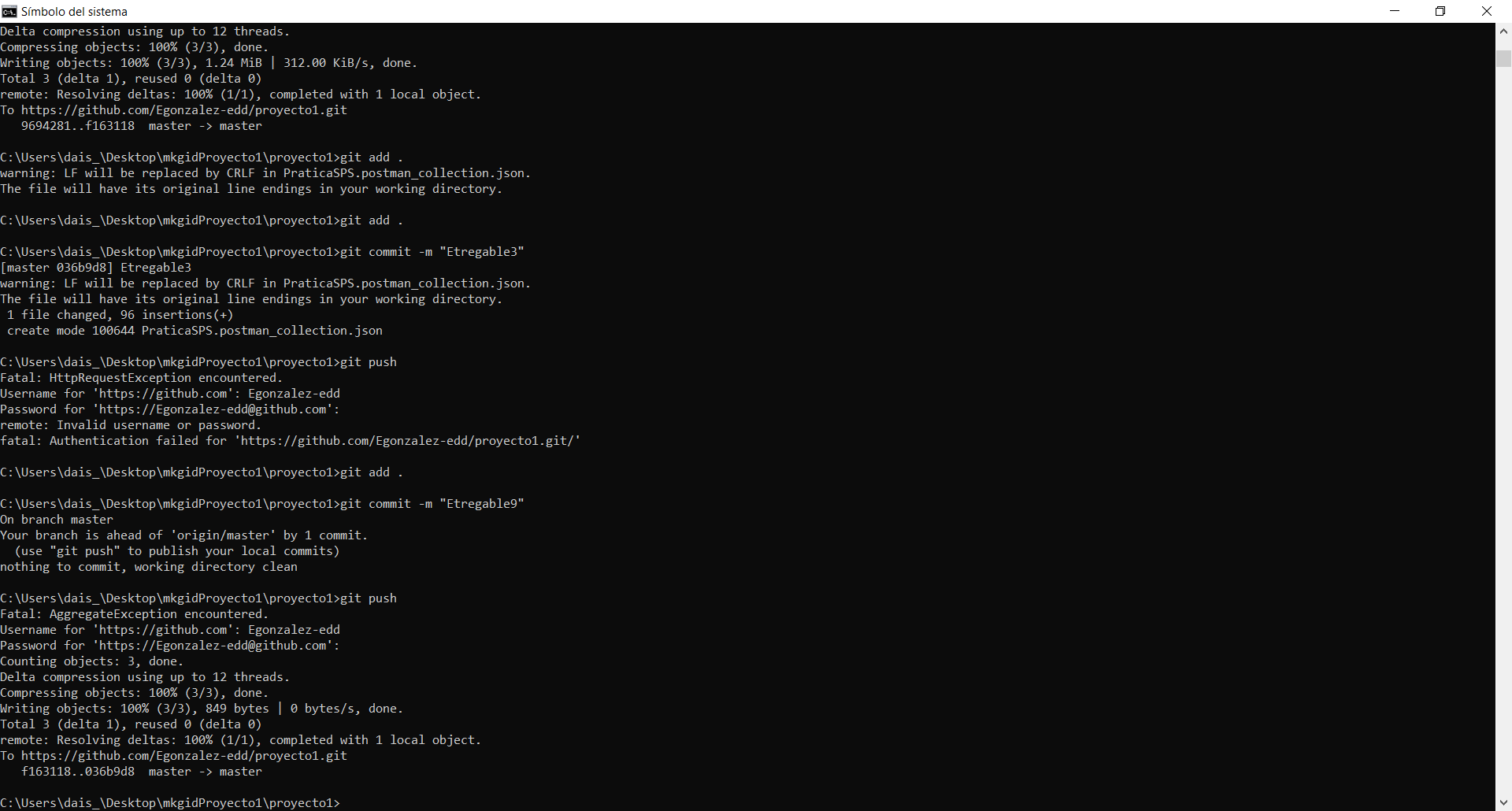
Ahora volvemos a correr el servicio de busqueda y podemos ver que el cliente con id 2 ya no existe y que se creo correctamente el cliente con id 100.



http://localhost:8080/ejemplo1/api/clientes?Content-Type=application/json

# GitHub





# Repositorio git.

<https://github.com/Egonzalez-edd/proyecto1.git>

