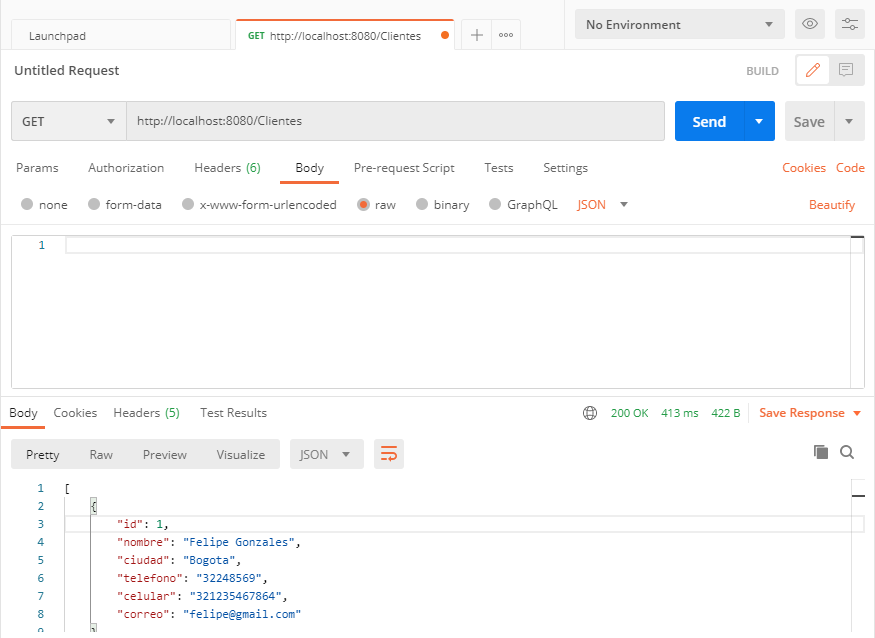
**Prueba de ingreso ST&T**

**Presentado por: Jhon Alexis Romero Hernandez**

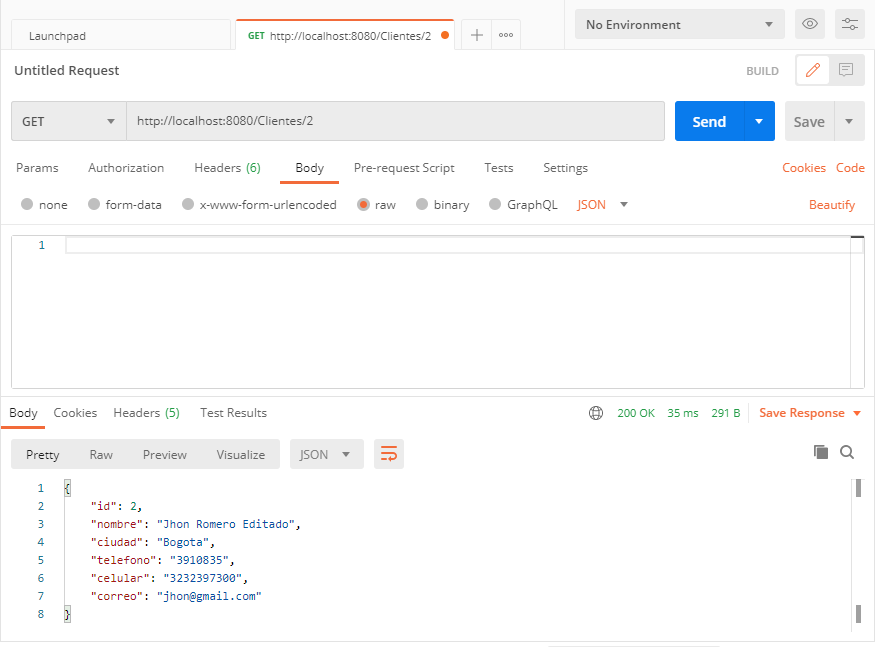
**¿Cómo probarlo?**

1. Métodos (REST): Recibe los parámetros necesarios para realizar un CRUD
2. Crear la base de datos, se encuentra en el archivo de Word explicación del código.
3. Correr el proyecto como Spring Boot App.
4. Una vez corra la aplicación se debe usar el programa PostMan para poder realizar pruebas correctamente.
5. Funciones:

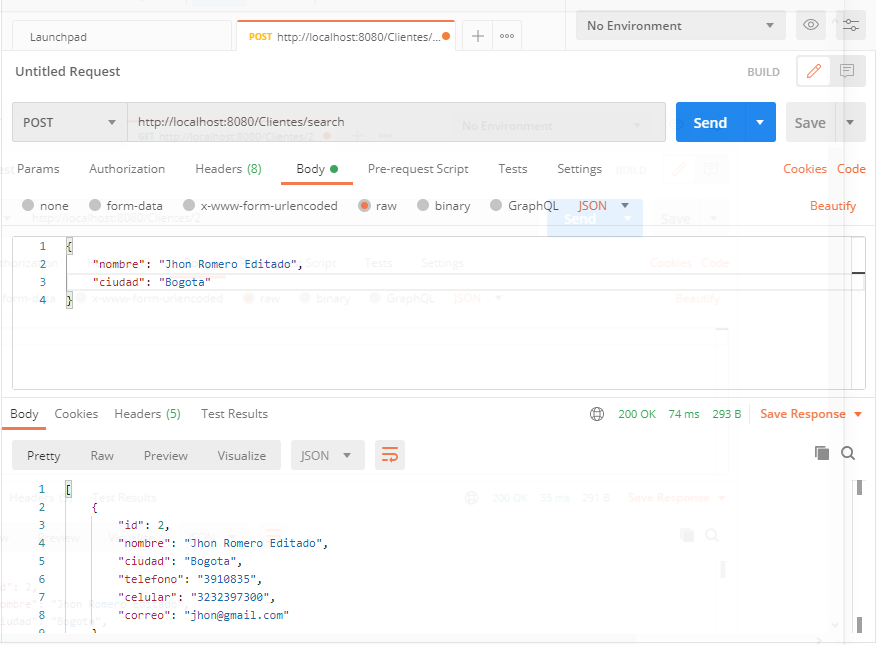
**getCliente**, usar la función GET de Postman, con el siguiente link, http://localhost:8080/Clientes. Esto obtendrá todos los clientes registrados en la base de datos



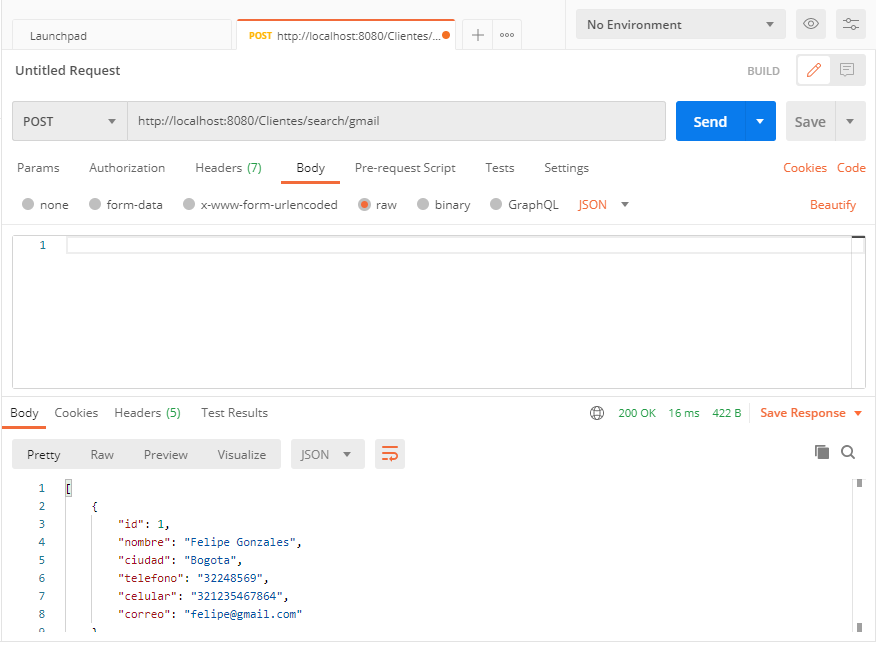
**findById**, usar la función GET de Postman, con el siguiente link, http://localhost:8080/Clientes/{aquí se debe ingresar el id del cliente a buscar sin corchetes}. Esto obtendrá la información del cliente a buscar registrado en la base de datos



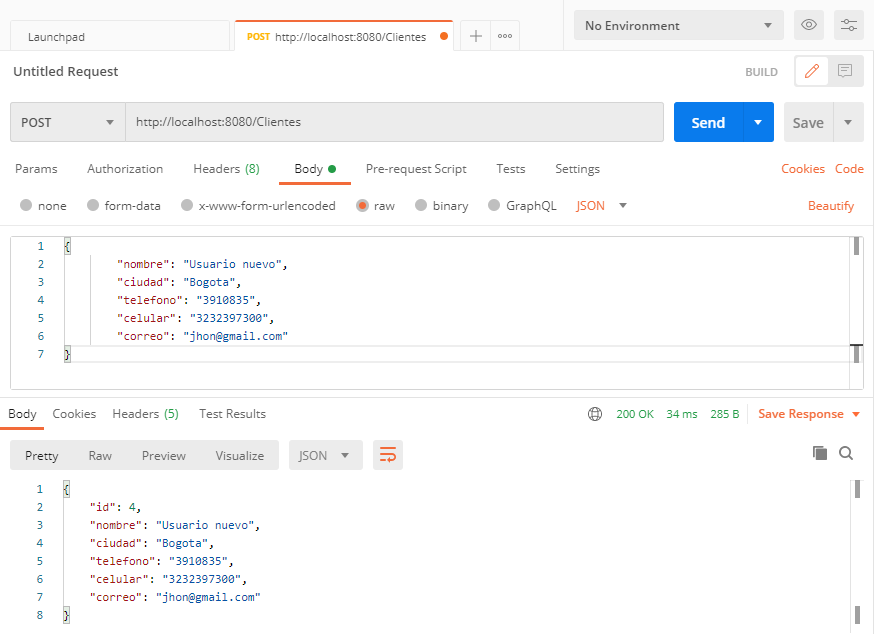
**findAdvancedCliente**, usar la función POST de Postman, con el siguiente link, http://localhost:8080/Clientes/search. Esto obtendrá la información de los clientes cuyo nombre y ciudad sean los ingresados en el apartado body cliente a buscar registrado en la base de datos



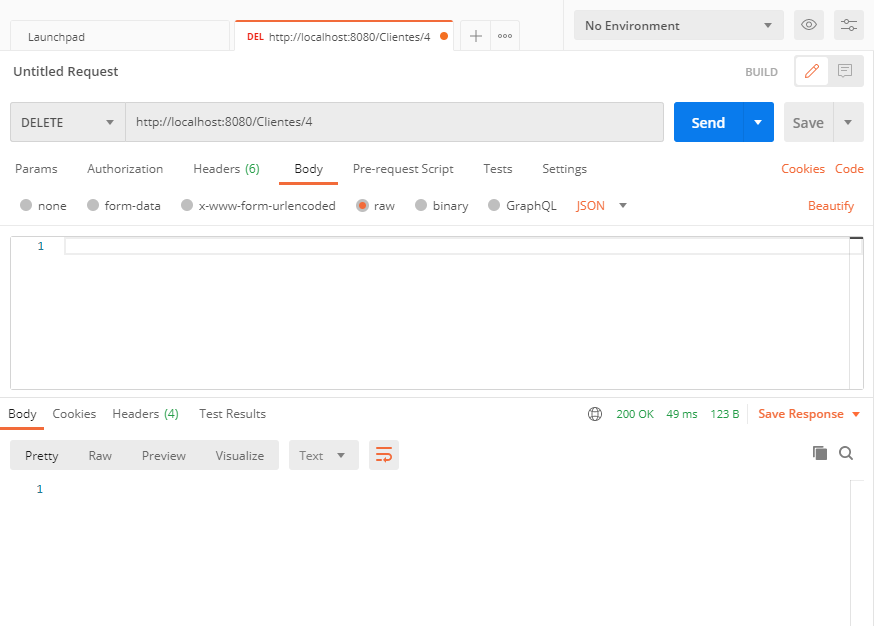
**findByCityAndGmailAdvanced**, usar la función POST de Postman, con el siguiente link, http://localhost:8080/Clientes/search/gmail. Esto obtendrá la información de los clientes que tengan como ciudad Bogotá y correo electrónico terminado en @gmail.com.



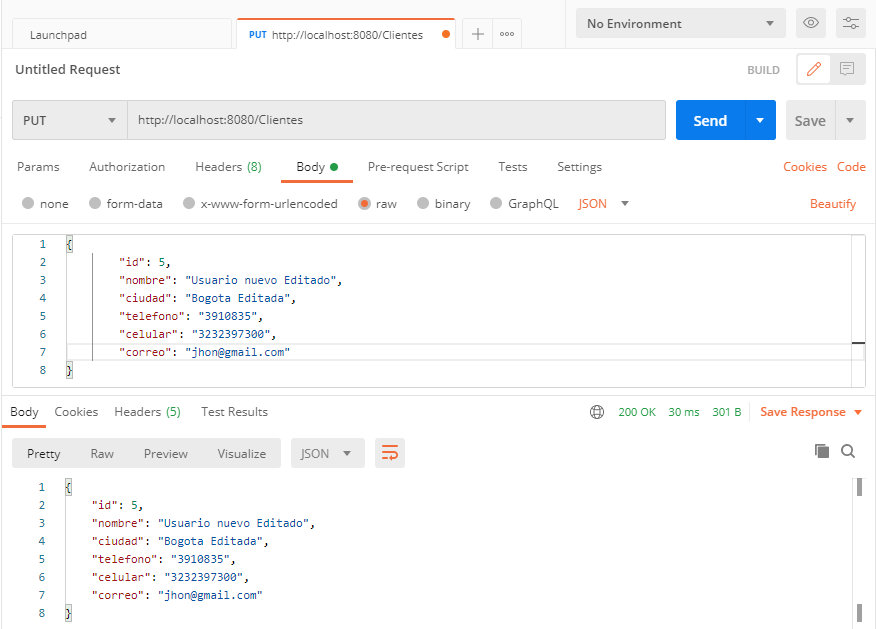
**createCliente**, usar la función POST de Postman, con el siguiente link, http://localhost:8080/Clientes. Esto creará un nuevo cliente según la información ingresada en el apartado body. Nota: Debe ser ingresado sin id, puesto que la columna id fue definida como autoincrementable.



**deleteCliente**, usar la función DELETE de Postman, con el siguiente link, http://localhost:8080/Clientes/{aquí se debe ingresar el id del cliente a eliminar sin corchetes}. Esto eliminará el cliente según la información del cliente a buscar registrado en la base de datos.

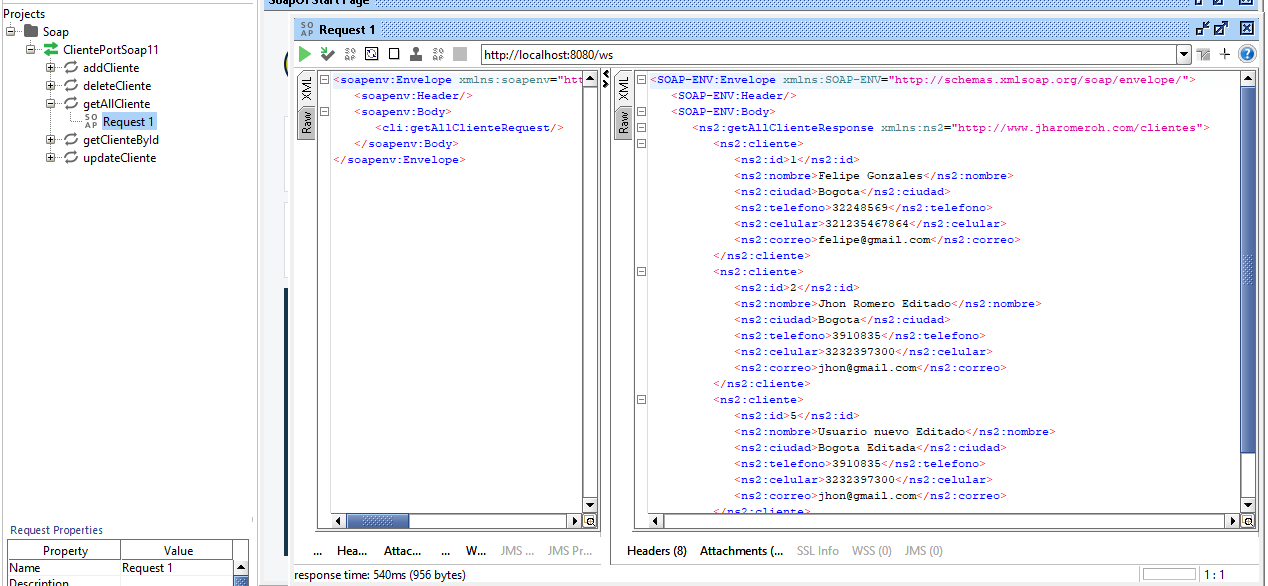


**updateCliente**, usar la función PUT de Postman, con el siguiente link, http://localhost:8080/Clientes. Esto editará un cliente según la información ingresada en el apartado body. Nota: Debe ser ingresado con id.

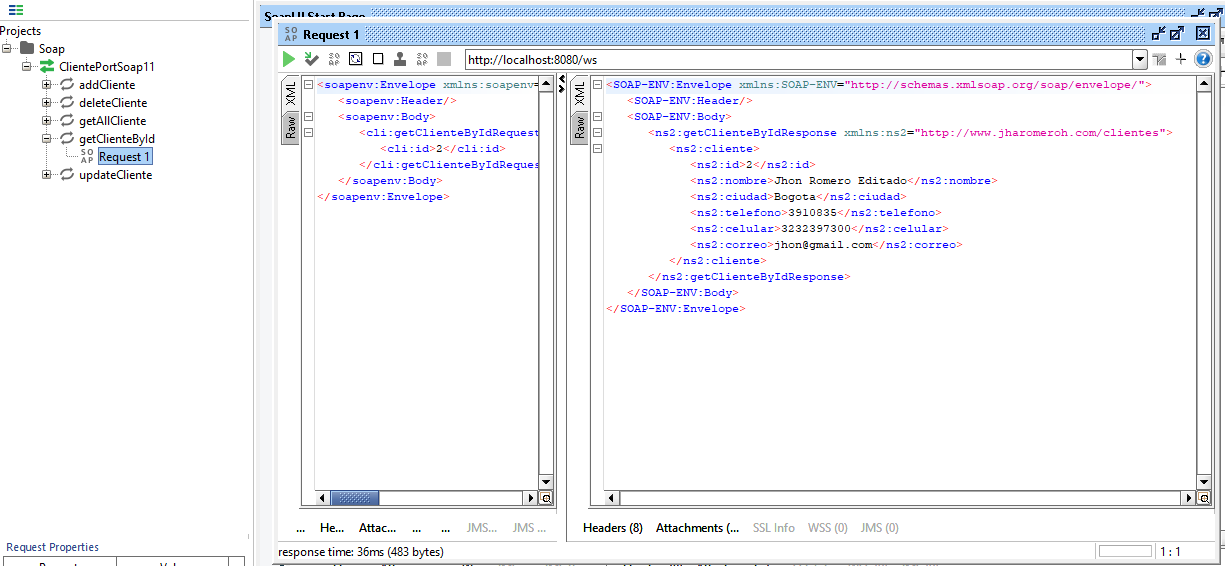


1. Métodos (SOAP): Recibe los parámetros necesarios para realizar un CRUD
2. Crear la base de datos, se encuentra en el archivo de Word explicación del código.
3. Correr el proyecto como Spring Boot App.
4. Una vez corra la aplicación se debe usar el programa SoapUI para poder realizar pruebas correctamente.
5. Crear un nuevo proyecto SOAP con cualquier nombre y la dirección localhost:8080/ws/cliente.wsdl.  
   Nota: En caso de no funcionar importar el archivo Soap-soapui-project.xml
6. Funciones:

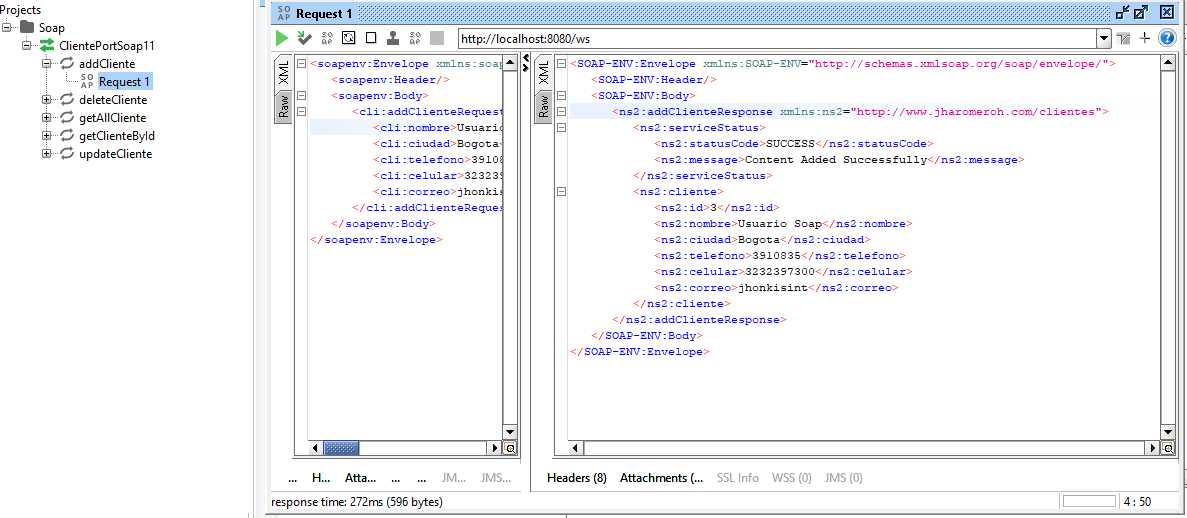
**getAllClienteRequest**, ejecturar la función con doble click en getAllCliente/Request1 y enviar como submit

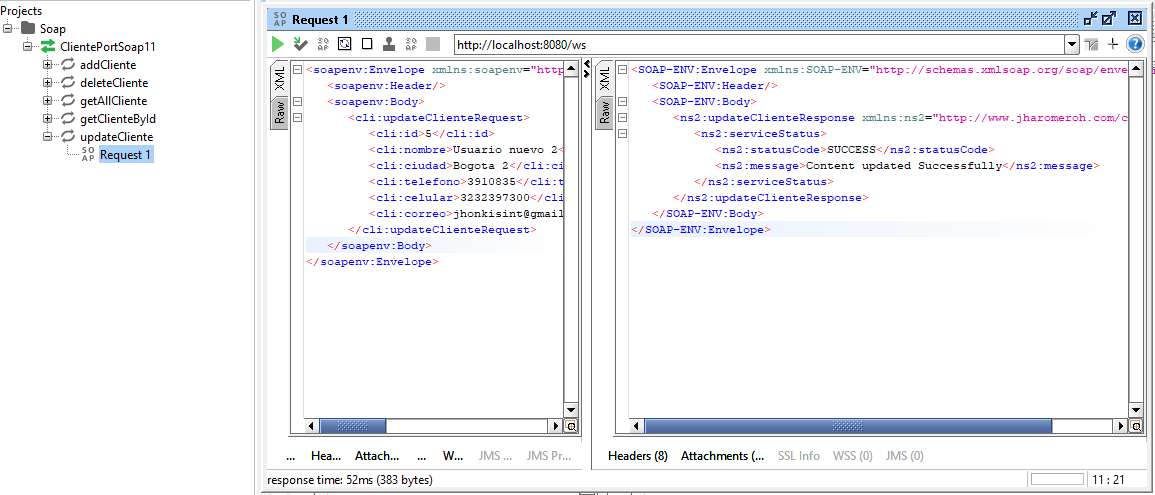


**getClienteByIdRequest**, ejecturar la función con doble click en **getClienteById**/Request1 en ingresar el id en el apartado id y enviar como submit



**addClienteRequest**, ejecturar la función con doble click en **addCliente**/Request1 en el cuerpo ingresar los datos necesarios para un nuevo usuario y enviar como submit.  
Nota: No se debe ingresar el id puesto que es autoincrementable en la BDD.

 **updateClienteRequest**, ejecturar la función con doble click en **updateCliente**/Request1 en el cuerpo ingresar los datos necesarios para un nuevo usuario y enviar como submit.  
Nota: Se debe ingresar el id.



**deleteClienteRequest**, ejecturar la función con doble click en **deleteCliente**/Request1 en el cuerpo ingresar el id a eliminar enviar como submit.  
