

Taller Arquitectura Cliente-Servidor: Sockets, WebServer y API Rest



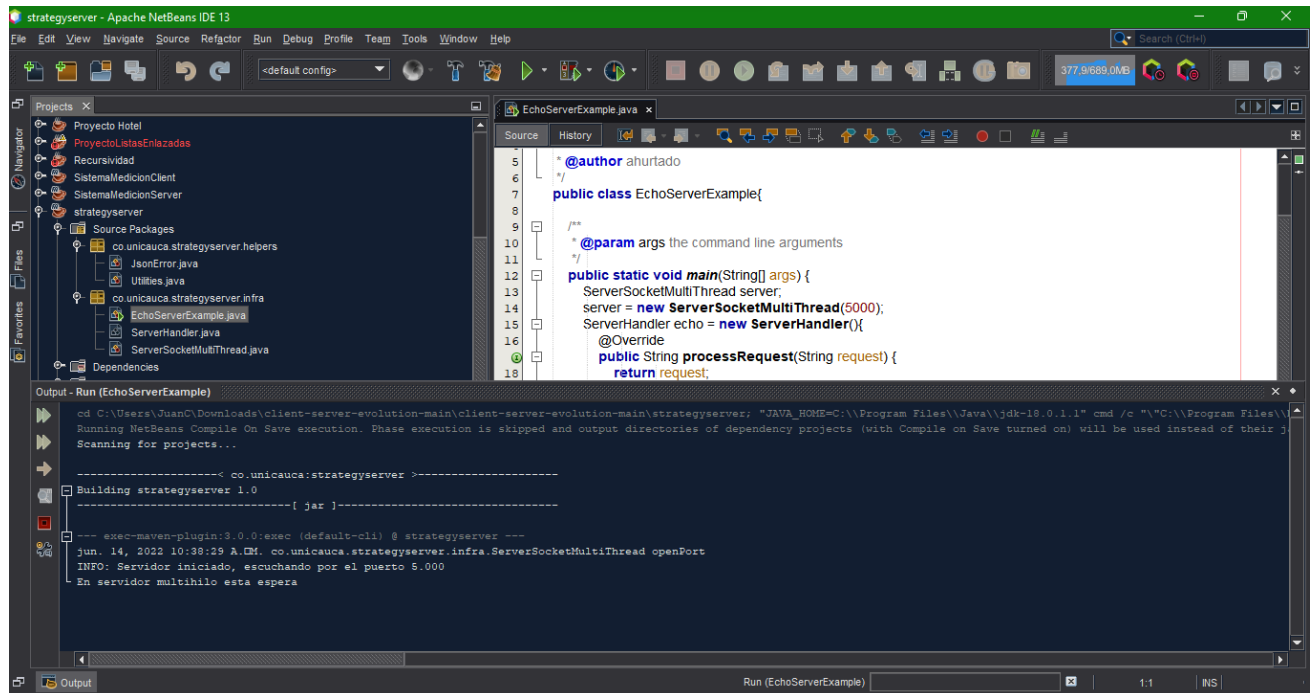
Presentado a:
Julio Ariel Hurtado

Presentado por:
Juan Carlos Neuta Montenegro

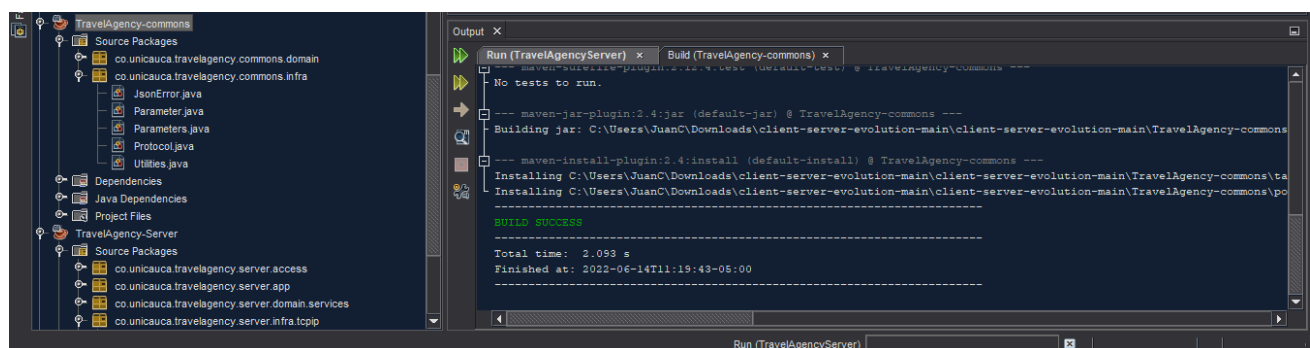
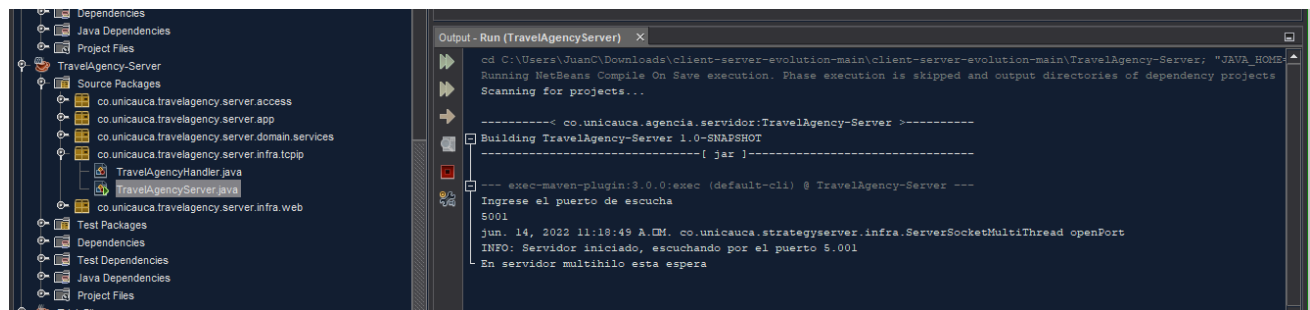
Facultad de Ingeniería Electrónica y Telecomunicaciones (FIET)
Programa de Ingeniería de sistemas
Popayán, junio del 2022

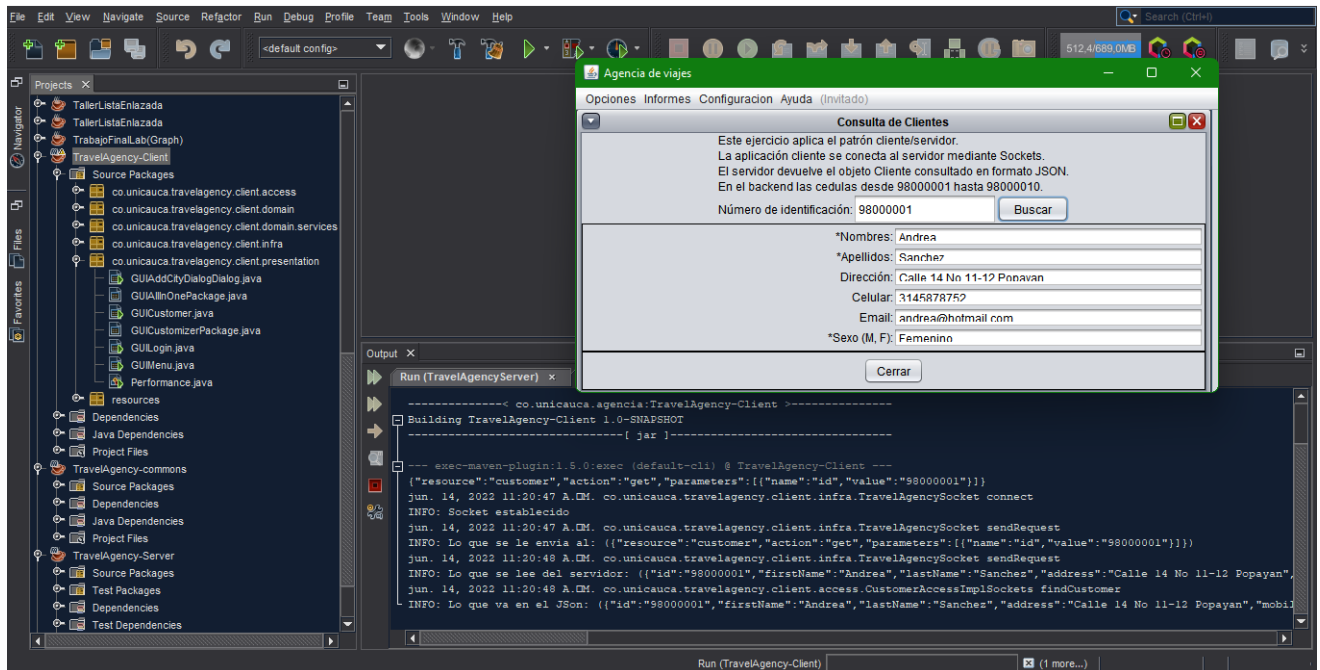
Tareas

1. Descargar, instalar y correr la clase EchoServerExample que viene en el proyecto strategyserver. Valor 0.5

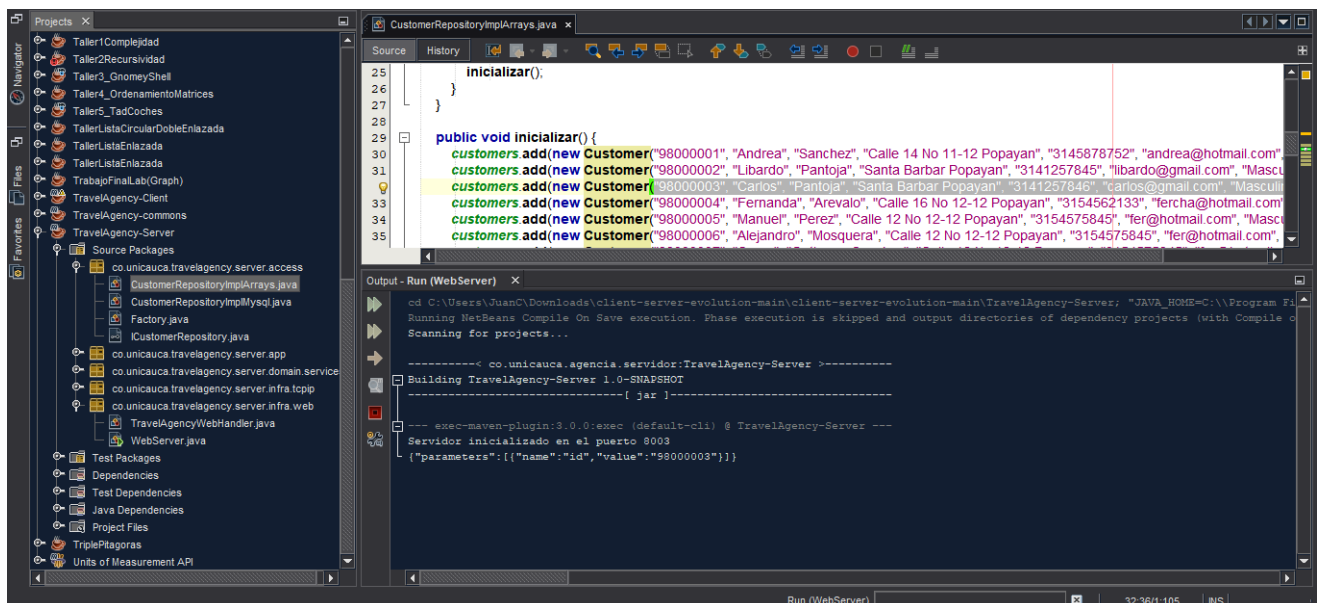


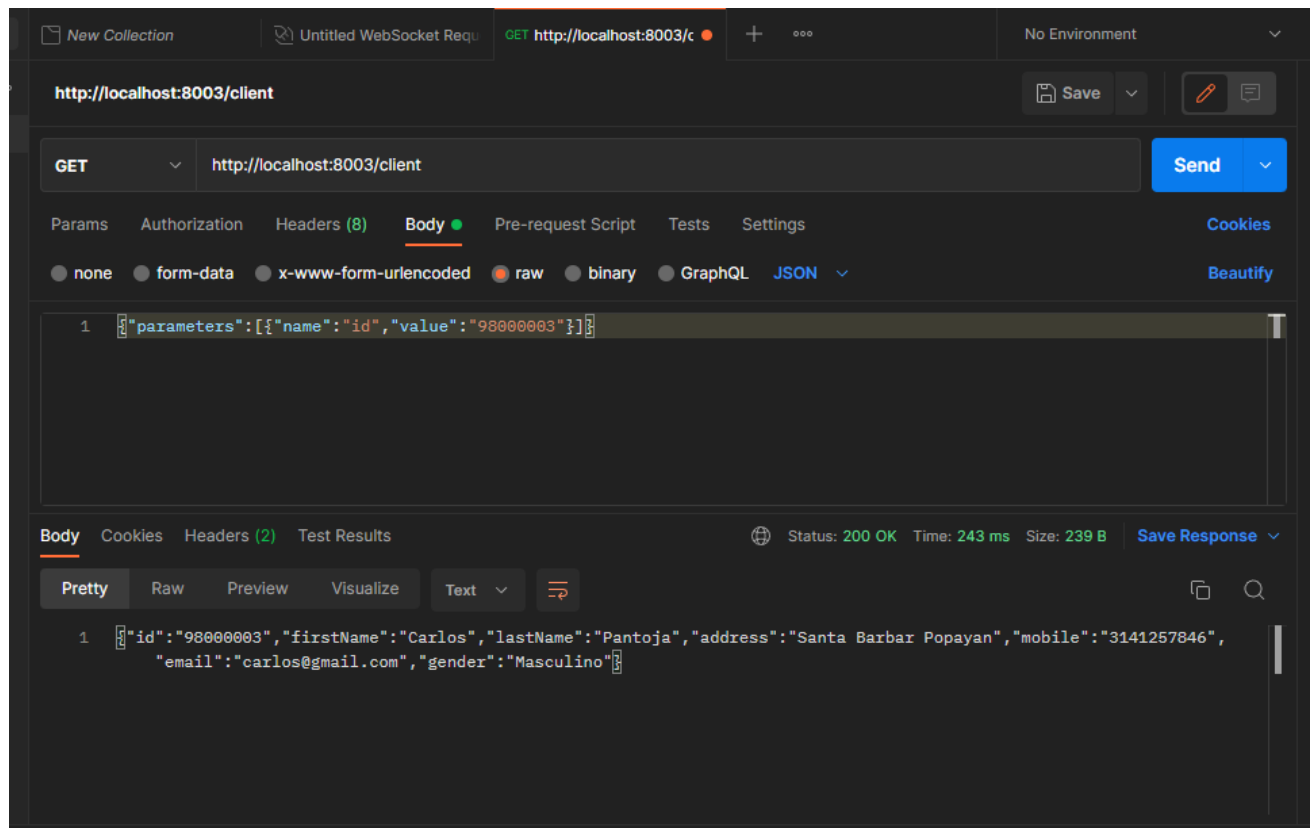
2. Descargar, instalar y correr el AgencyTravelServer como servidor tcp/ip y el AgencyTravel-Client (infra.tcpip). Valor 0.5



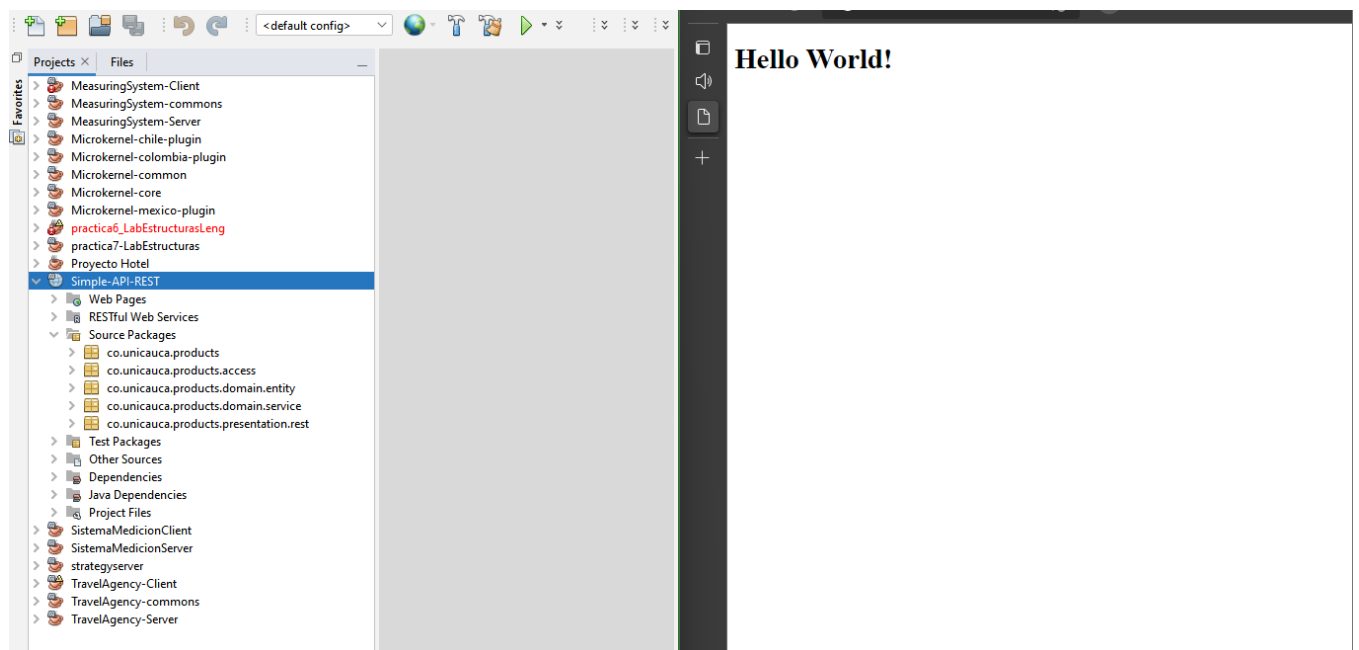


3. Correr el Web Server(infra.web) que vienen en el mismo proyecto AgencyTravelServer. Usar un cliente postman para hacer la consulta, recuerden que el protocolo cambio y ahora sólo van los parámetros. Ver la consulta postman. Valor 0.5





4. Realizar el taller de la API-Rest y probar las consultas a través de postman o un Jersey Client. Siguen el paso a paso del taller. Valor 1.5



Listar todos los productos

The screenshot shows the Postman interface for a GET request to `http://localhost:8080/Simple-API-REST/product-service/products/`. The request is successful with a status of 200 OK, a time of 26 ms, and a size of 392 B. The response body is displayed in JSON format, showing a list of three products:

```
{
  "id": 1,
  "name": "Tv samsung",
  "price": 200000.0
},
{
  "id": 2,
  "name": "Tv lg",
  "price": 300000.0
},
{
  "id": 1,
  "name": "Tablet asus RESD-TD-34",
  "price": 400000.0
}
```

Listar un producto en particular, en este ejemplo con id=1

The screenshot shows the Postman interface for a GET request to `http://localhost:8080/Simple-API-REST/product-service/products/1`. The request is successful with a status of 200 OK, a time of 19 ms, and a size of 290 B. The response body is displayed in JSON format, showing the details of the product with ID 1:

```
{
  "id": 1,
  "name": "Tv samsung",
  "price": 200000.0
}
```

Crear un producto.

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/Simple-API-REST/product-service/products/`
- Method:** `POST`
- Body:** `{ "id": 5, "name": "Tv Xiaomi", "price": 250000.0 }`
- Response:** `{ "ok": "true", "mensaje": "Producto creado", "errores": "" }`
- Status:** `200 OK`, **Time:** `15 ms`, **Size:** `294 B`

Eliminar un producto con id=2

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/Simple-API-REST/product-service/products/2`
- Method:** `DELETE`
- Body:** This request does not have a body
- Response:** `{ "ok": "true", "mensaje": "Producto borrado", "errores": "" }`
- Status:** `200 OK`, **Time:** `6 ms`, **Size:** `295 B`

Editar un producto

The screenshot shows a REST client interface with a dark theme. At the top, the URL bar displays `http://localhost:8080/Simple-API-REST/product-service/products/`. Below the URL bar, the HTTP method is set to `PUT`. The request body is a JSON object: `{ "id": 5, "name": "Cell Xiaomi", "price": 250000.0 }`. The response status is `200 OK` with a time of `13 ms` and a size of `298 B`. The response body is a JSON object: `{ "ok": "true", "mensaje": "Producto modificado", "errores": "" }`.

http://localhost:8080/Simple-API-REST/product-service/products/ Save

PUT http://localhost:8080/Simple-API-REST/product-service/products/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 { "id": 5, "name": "Cell Xiaomi", "price": 250000.0 }
```

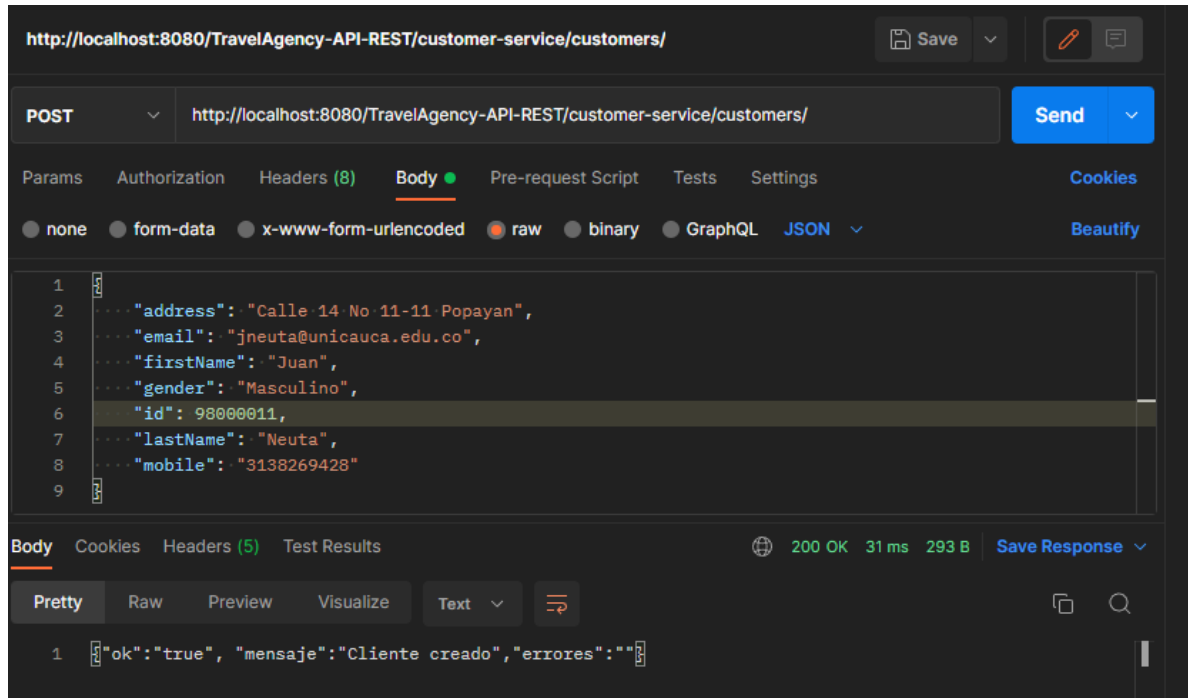
Body Cookies Headers (5) Test Results Status: 200 OK Time: 13 ms Size: 298 B Save Response

Pretty Raw Preview Visualize Text

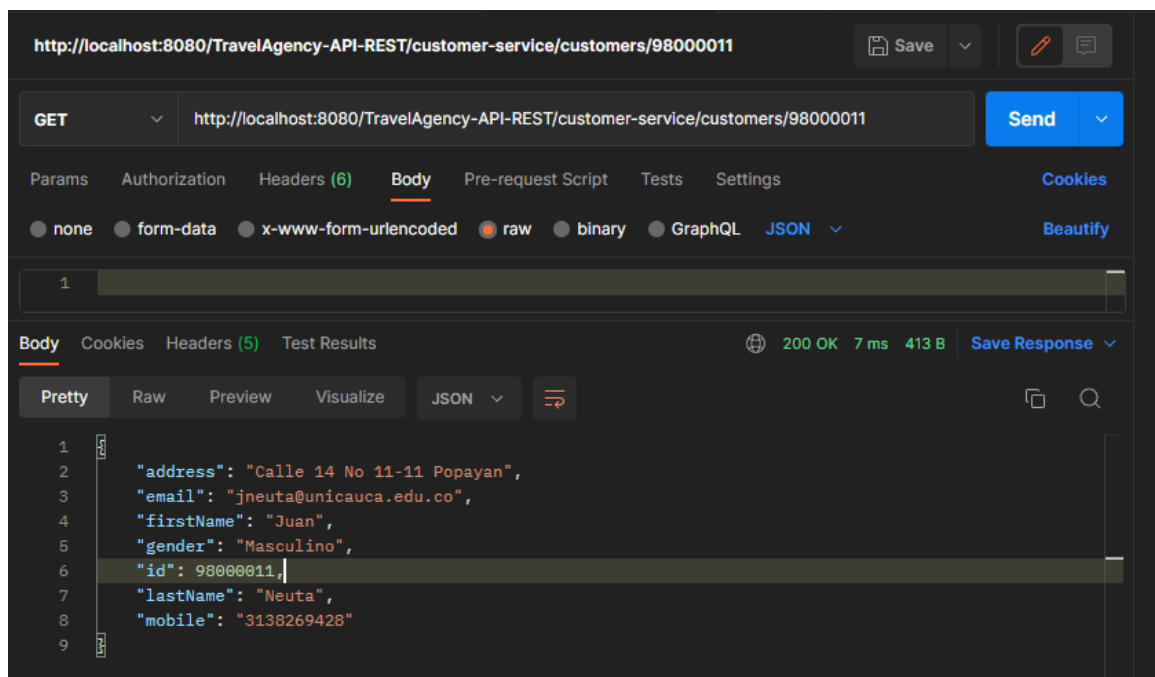
```
1 { "ok": "true", "mensaje": "Producto modificado", "errores": "" }
```

5. Hacer una API Rest para el AgencyTravelServer. Probarla desde un Jersey Client o a través de postman. Valor Valor 2.0.

Crear o Registrar un cliente de la agencia de viaje



Buscar un cliente de la agencia de viaje



Link Repositorio de github :

<https://github.com/JuanCNeuta/Taller-Arquitectura-Cliente-Servidor-Sockets-WebServer-y-API-Rest>