

Taller Arquitectura Cliente-Servidor, WebServer y API Rest



Universidad
del Cauca
Vigilada Mineducación

Ingeniería de software II

Presentado por:

WILLIAM CAICEDO MAGIN

Profesor:

JULIO ARIEL HURTADO

Universidad del Cauca

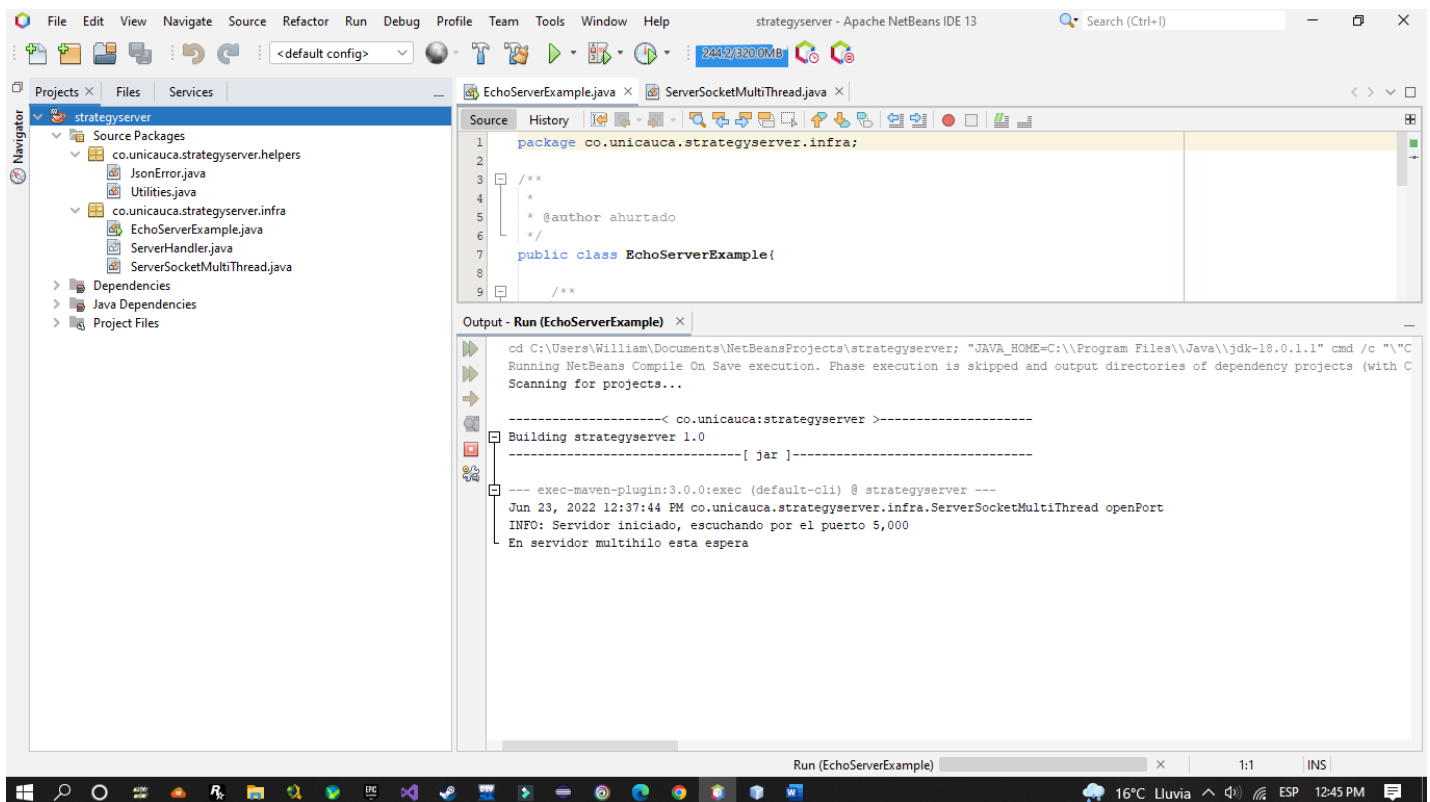
Facultad de Ingeniería Electrónica y Telecomunicaciones

Ingeniería de sistemas

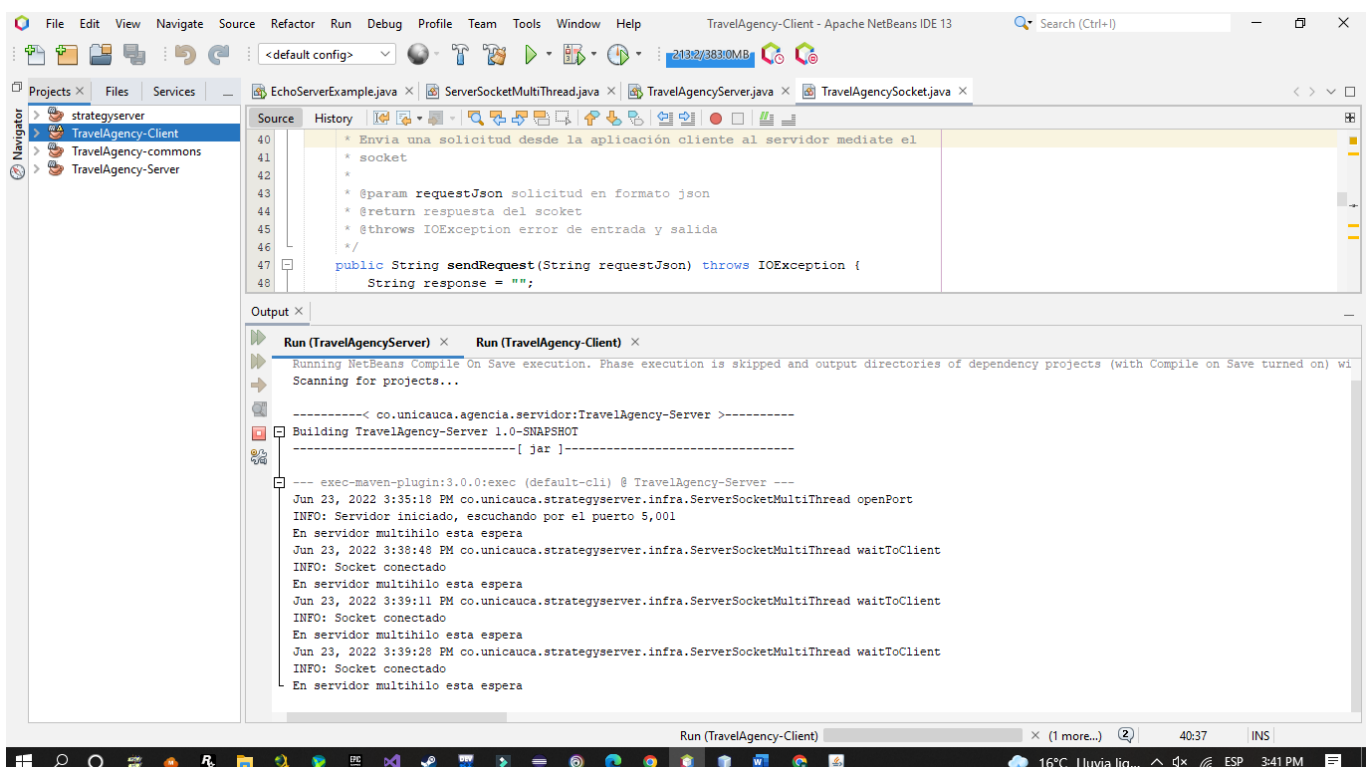
Popayán, junio 24 de 2022

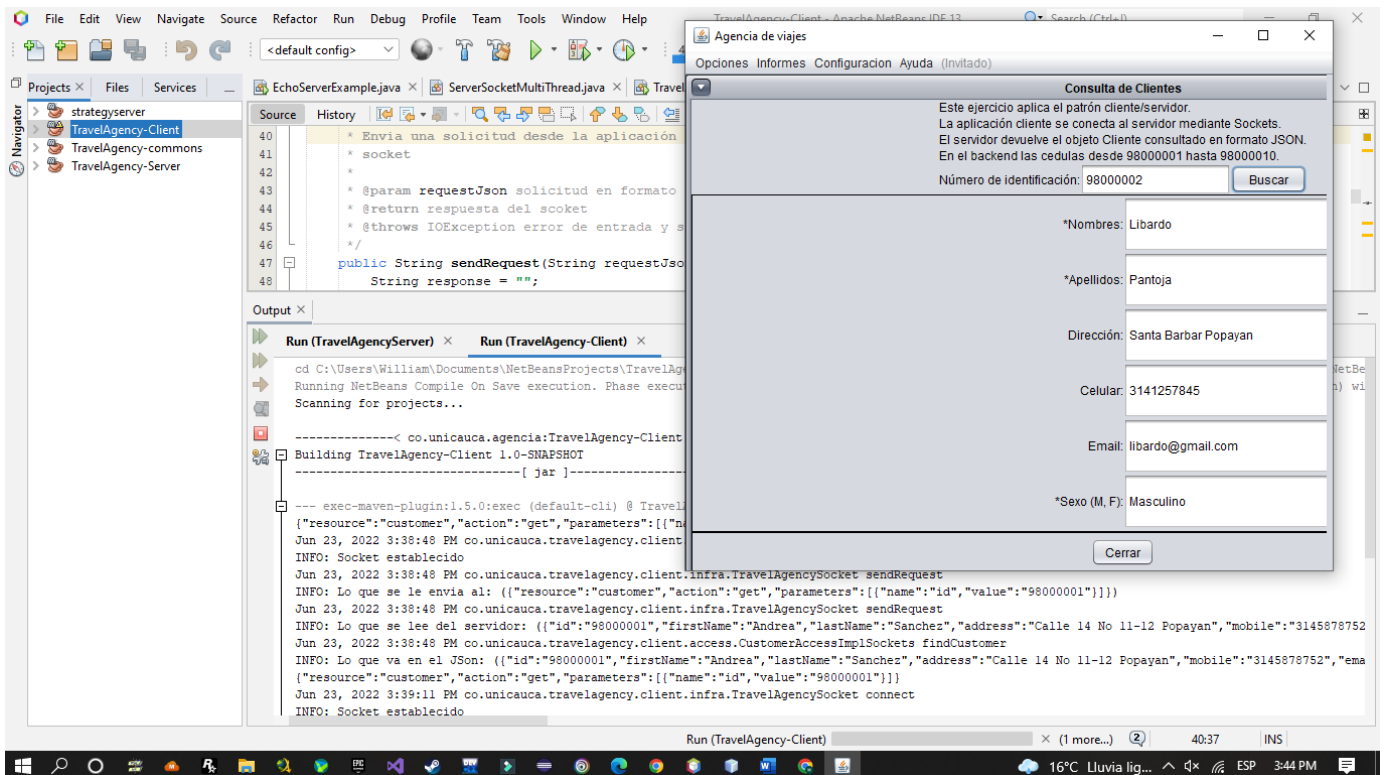
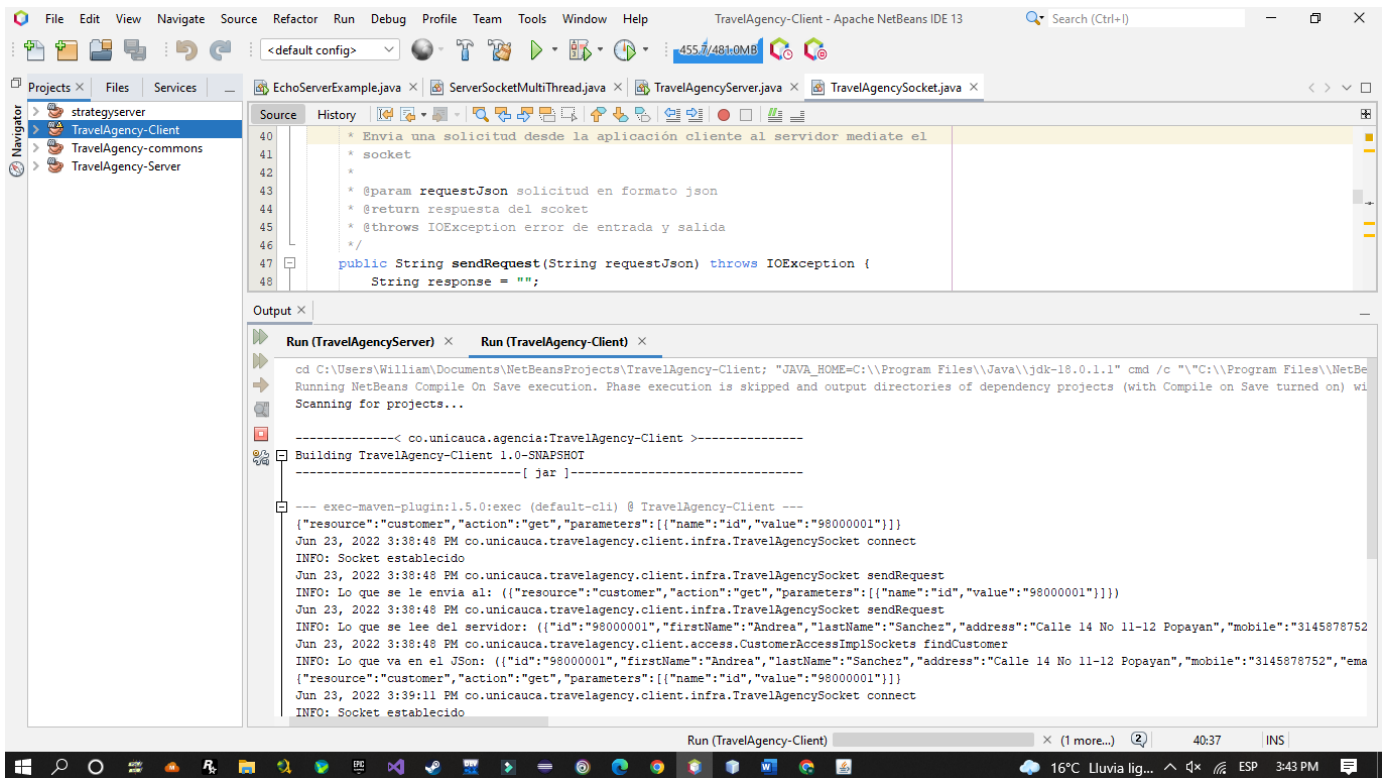
TALLER.

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

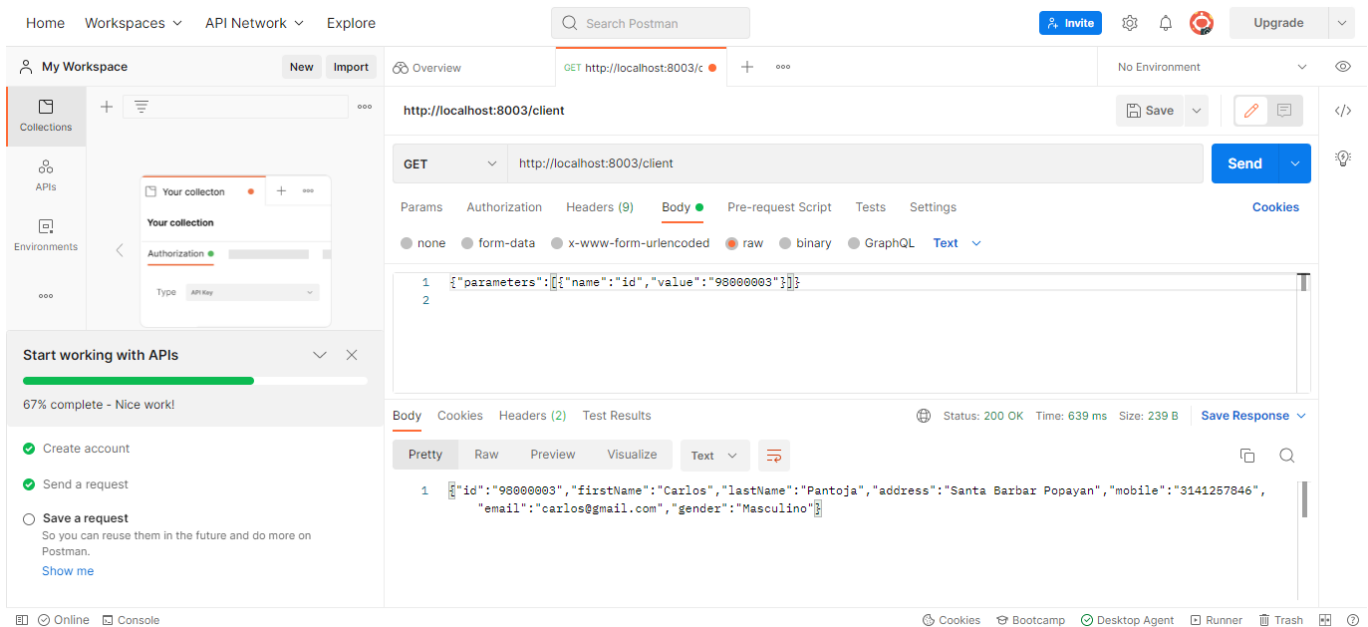
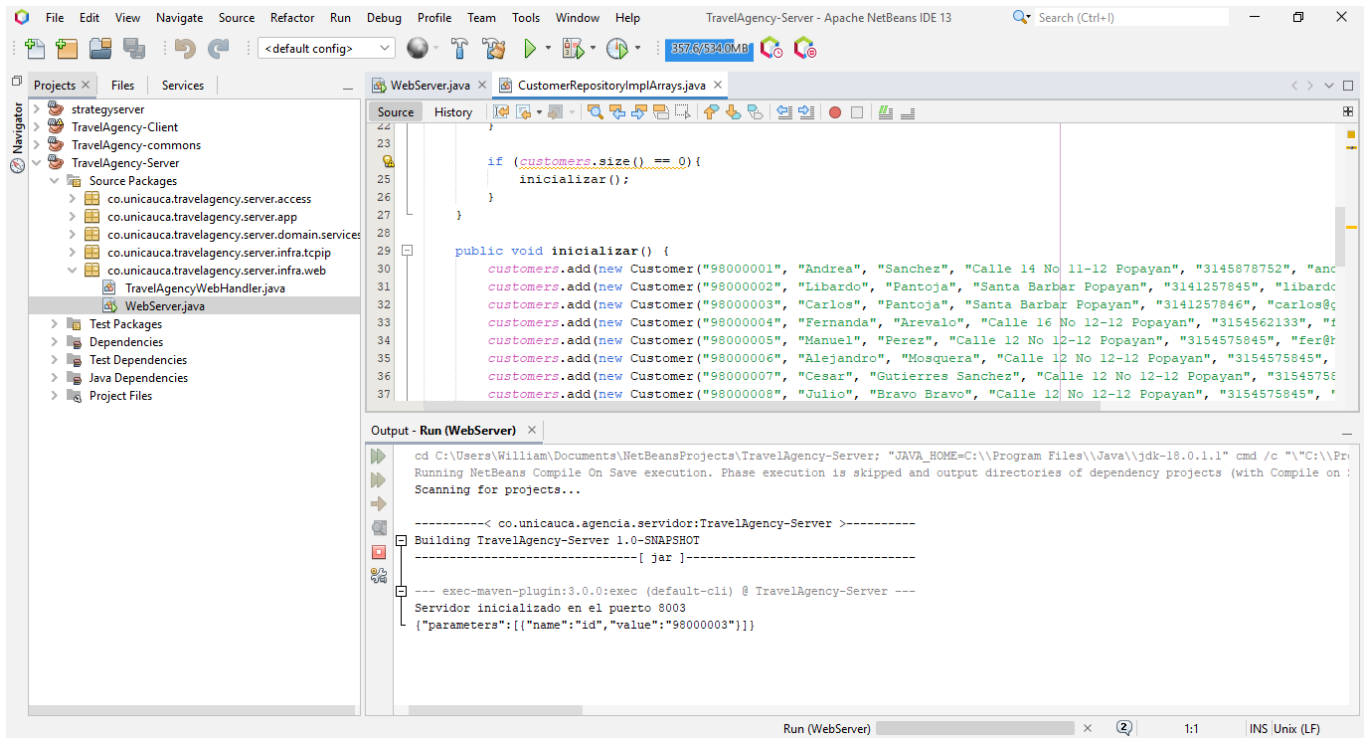


2. Descargar, instalar y correr el AgencyTravelServer como servidor tcp/ip y el AgencyTravelClient (infra.tcpip).

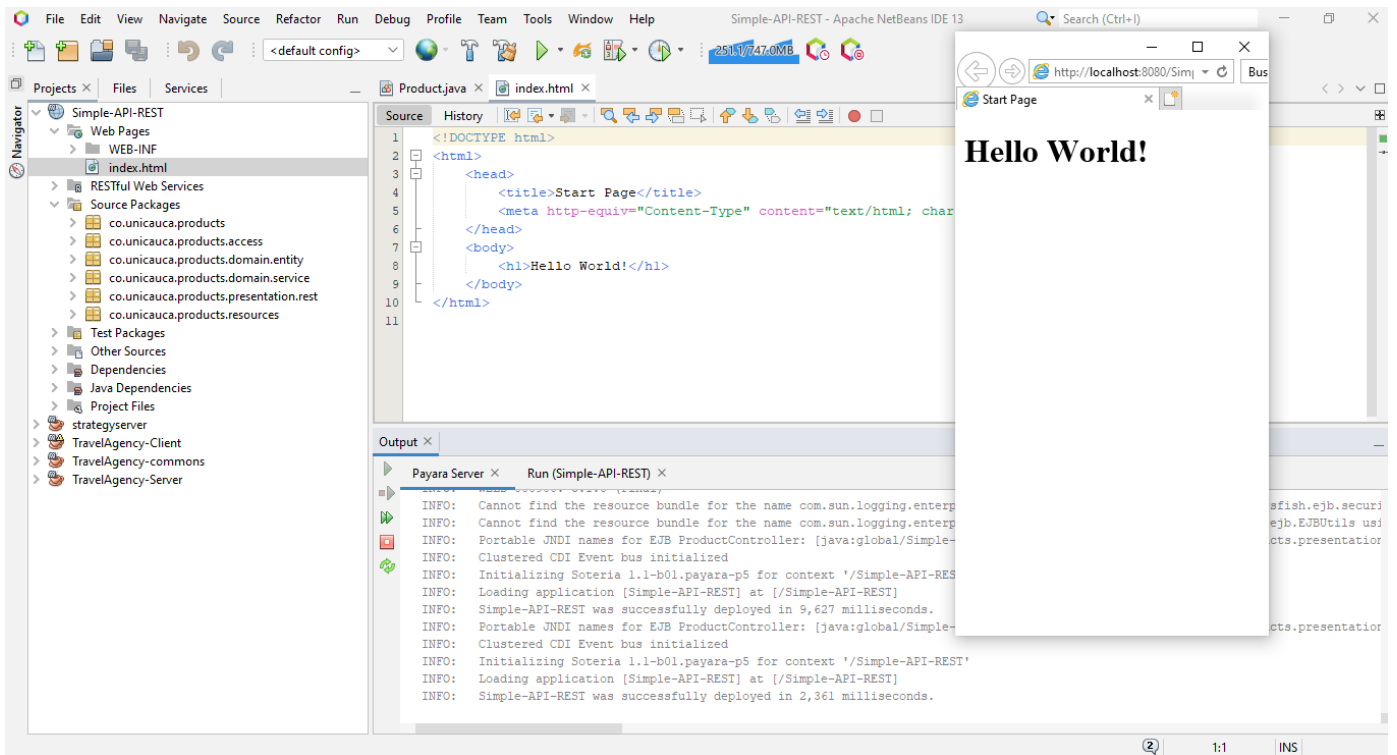




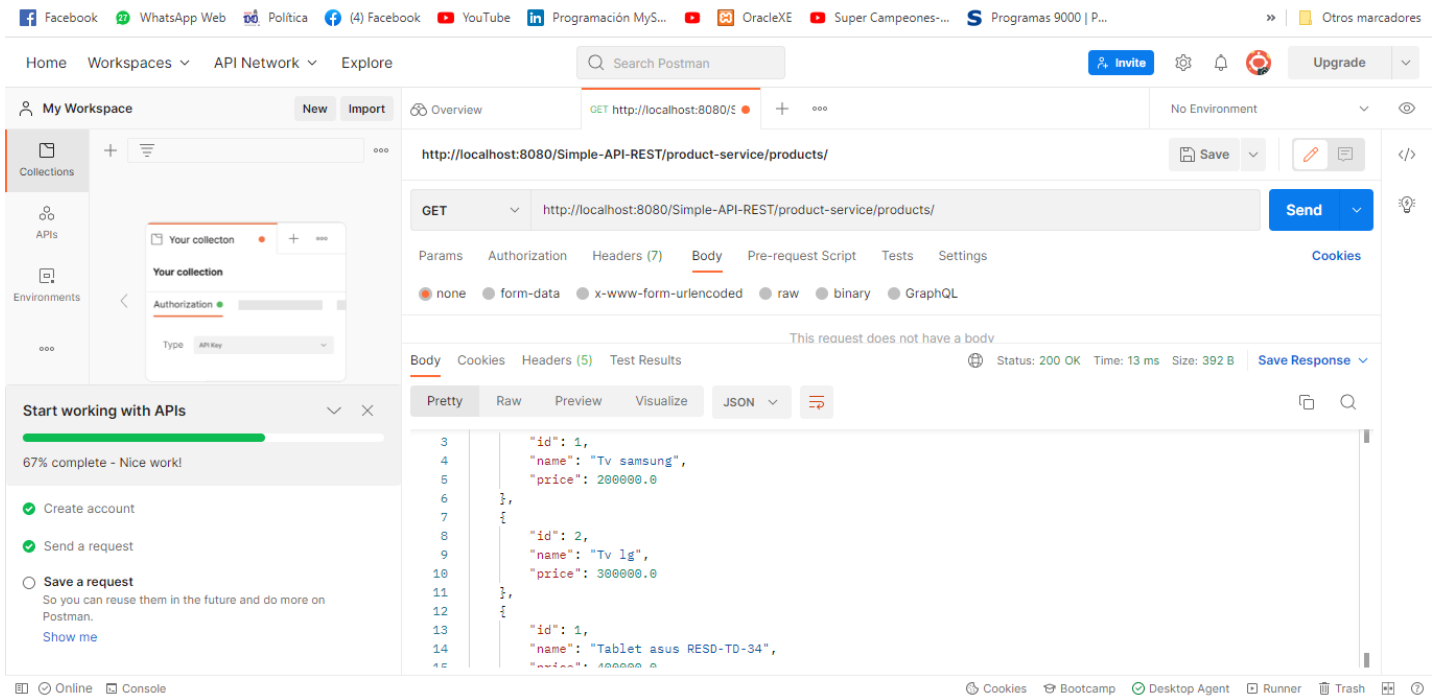
3. Correr el Web Server(infra.web) que vienen en el mismo proyecto AgencyTravelServer. Usar un cliente postman para hacer la consulta, recuerden que el protocolo cambio y ahora solo van los parámetros.



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



- Listar todos los productos mediante postman.



- **Listar un producto en particular, mediante postman.**

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/Simple-API-REST/product-service/products/1`. The request is successful, returning a 200 OK status. The response body is displayed in JSON format:

```
1 {
2   "id": 1,
3   "name": "Tv samsung",
4   "price": 200000.0
5 }
```

The left sidebar shows the 'My Workspace' section with a 'Your collection' and an 'Authorization' tab. The bottom status bar indicates the request was successful with a 200 OK status, 16 ms time, and 290 B size.

- **Crear un producto mediante postman.**

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/Simple-API-REST/product-service/products/`. The request is successful, returning a 200 OK status. The response body is displayed in JSON format:

```
1 {
2   "id": 50,
3   "name": "Nevera Lg",
4   "price": 670000.0
5 }
```

The left sidebar shows the 'My Workspace' section with a 'Your collection' and an 'Authorization' tab. The bottom status bar indicates the request was successful with a 200 OK status, 14 ms time, and 293 B size.

- Eliminar un producto, mediante postman.

The screenshot shows the Postman interface with a DELETE request to `http://localhost:8080/Simple-API-REST/product-service/products/2`. The request is in the 'Body' tab, and the response is a JSON object: `{\"ok\":true,\"mensaje\":\"Producto borrado\",\"errores\":\"\"}`. The status is 200 OK, time is 17 ms, and size is 294 B.

- Editar un producto, mediante postman.

The screenshot shows the Postman interface with a PUT request to `http://localhost:8080/Simple-API-REST/product-service/products/`. The request is in the 'Body' tab, and the response is a JSON object: `{\"ok\":true,\"mensaje\":\"Producto modificado\",\"errores\":\"\"}`. The status is 200 OK, time is 11 ms, and size is 297 B.

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

➤ Link github: <https://github.com/Cwilliam97/APITravelAgency.git>

- Listar todos los clientes mediante postman.

The screenshot shows the Postman interface with a workspace named 'My Workspace'. A collection of requests is visible on the left, all pointing to 'http://localhost:8080/TravelAgency-'. The selected request is a GET request to 'http://localhost:8080/TravelAgency-API/travelagency-service/customers/'. The response is a 200 OK status with a response time of 16 ms and a size of 1.81 KB. The response body is displayed in JSON format, showing an array of two customer objects:

```
11 {
12   "address": "Santa Barbar Popayan",
13   "email": "libardo@gmail.com",
14   "firstName": "Libardo",
15   "gender": "Masculino",
16   "id": 2,
17   "lastName": "Pantoja",
18   "mobile": "3141257845"
19 },
20 {
21   "address": "Santa Barbar Popayan",
22   "email": "carlos@gmail.com",
23   "firstName": "Carlos",
24   "gender": "Masculino",
25   "id": 3,
26   "lastName": "Pantoja",
27   "mobile": "3141257845"
28 }
```

- Listar un cliente en particular, mediante postman.

The screenshot shows the Postman interface with the same workspace. The selected request is a GET request to 'http://localhost:8080/TravelAgency-API/travelagency-service/customers/1'. The response is a 200 OK status with a response time of 15 ms and a size of 405 B. The response body is displayed in JSON format, showing a single customer object:

```
1 {
2   "address": "Calle 14 No 11-12 Popayan",
3   "email": "andrea@hotmail.com",
4   "firstName": "Andrea",
5   "gender": "Femenino",
6   "id": 1,
7   "lastName": "Sanchez",
8   "mobile": "3145878752"
9 }
```


- Crear un producto mediante postman.

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The left sidebar lists various collections and environments. The main panel displays a POST request to 'http://localhost:8003/client'. The request body is a JSON object representing a customer:

```
{  "id": 50,  "firstName": "William",  "lastName": "Caicedo",  "address": "Carrera 1",  "mobile": "12345678",  "email": "wcaicedom@gmail.com",  "gender": "Masculino"}
```

. The response is a JSON object:

```
{  "ok": "true",  "mensaje": "Cliente creado",  "errores": ""}
```

. The status is 200 OK, and the response is saved.

- Eliminar un cliente, mediante postman.

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The left sidebar lists various collections and environments. The main panel displays a DELETE request to 'http://localhost:8080/TravelAgency-API-REST/travelagency-service/customers/1'. The response is a JSON object:

```
{  "ok": "true",  "mensaje": "Cliente borrado",  "errores": ""}
```

. The status is 200 OK, and the response is saved.

- **Editar un producto, mediante postman.**

The screenshot displays the Postman API client interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore'. A search bar labeled 'Search Postman' is present. On the right, there are buttons for 'invite', 'Upgrade', and a dropdown menu.

The left sidebar shows the 'My Workspace' section with a 'New' button and an 'Import' button. Below these are icons for 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'.

The main workspace area is divided into two panes. The top pane shows a list of API requests under the 'Today' tab, all pointing to 'http://localhost:8080/TravelAgency-'. The bottom pane shows a detailed view of a PUT request to 'http://localhost:8003/client'. The request is configured with the following details:

- Method:** PUT
- URL:** http://localhost:8080/TravelAgency-API/travelagency-service/customers/
- Body:** A JSON object representing a customer:

```
{  "id": 50,  "firstName": "Lucas",  "lastName": "Perez",  "address": "Carrera 1",  "mobile": "12345678",  "email": "wcaicedom@gmail.com",  "gender": "Masculino"}
```
- Headers:** 9 headers are listed.
- Body Type:** JSON
- Status:** 200 OK
- Time:** 27 ms
- Size:** 296 B

The bottom pane shows the response body in 'Pretty' format, which is a JSON object:

```
{  "ok": "true",  "mensaje": "Cliente modificado",  "errores": ""}
```

The bottom status bar includes icons for 'Online', 'Console', 'Cookies', 'Bootcamp', 'Desktop Agent', 'Runner', 'Trash', and a help icon.