

Taller 2 Cliente-Servidor, API-REST

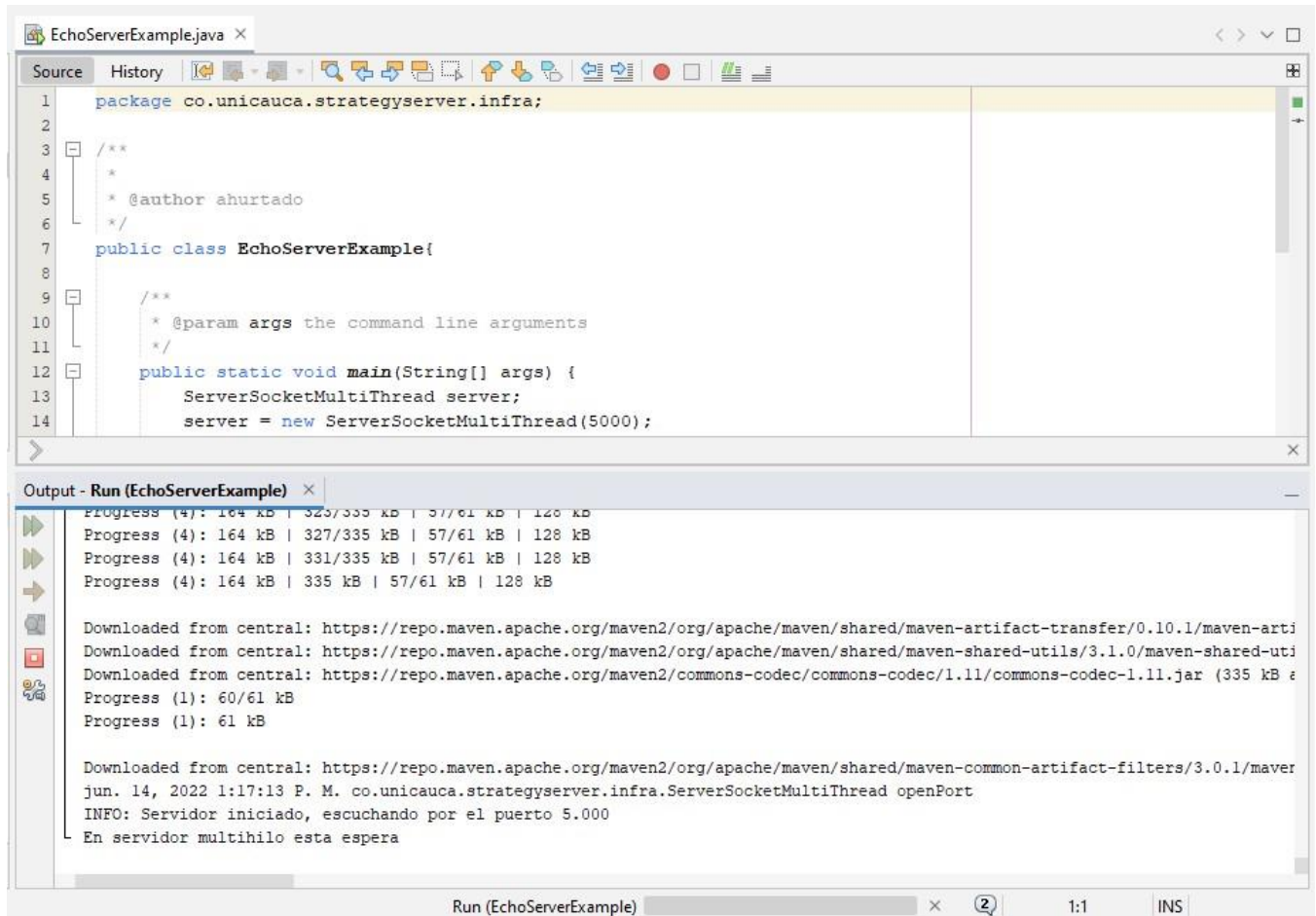
Ingeniería de Software II

Ing. Julio Ariel Hurtado

Juan Sebastián Osorio Aguilera

Popayán 2022

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



The screenshot shows an IDE window with two tabs: 'EchoServerExample.java' and 'Output - Run (EchoServerExample)'. The 'EchoServerExample.java' tab is active, displaying the following code:

```
1 package co.unicauca.strategyserver.infra;
2
3 /**
4  *
5  * @author ahurtado
6  */
7 public class EchoServerExample{
8
9     /**
10      * @param args the command line arguments
11      */
12     public static void main(String[] args) {
13         ServerSocketMultiThread server;
14         server = new ServerSocketMultiThread(5000);
```

The 'Output - Run (EchoServerExample)' tab is also active, showing the following output:

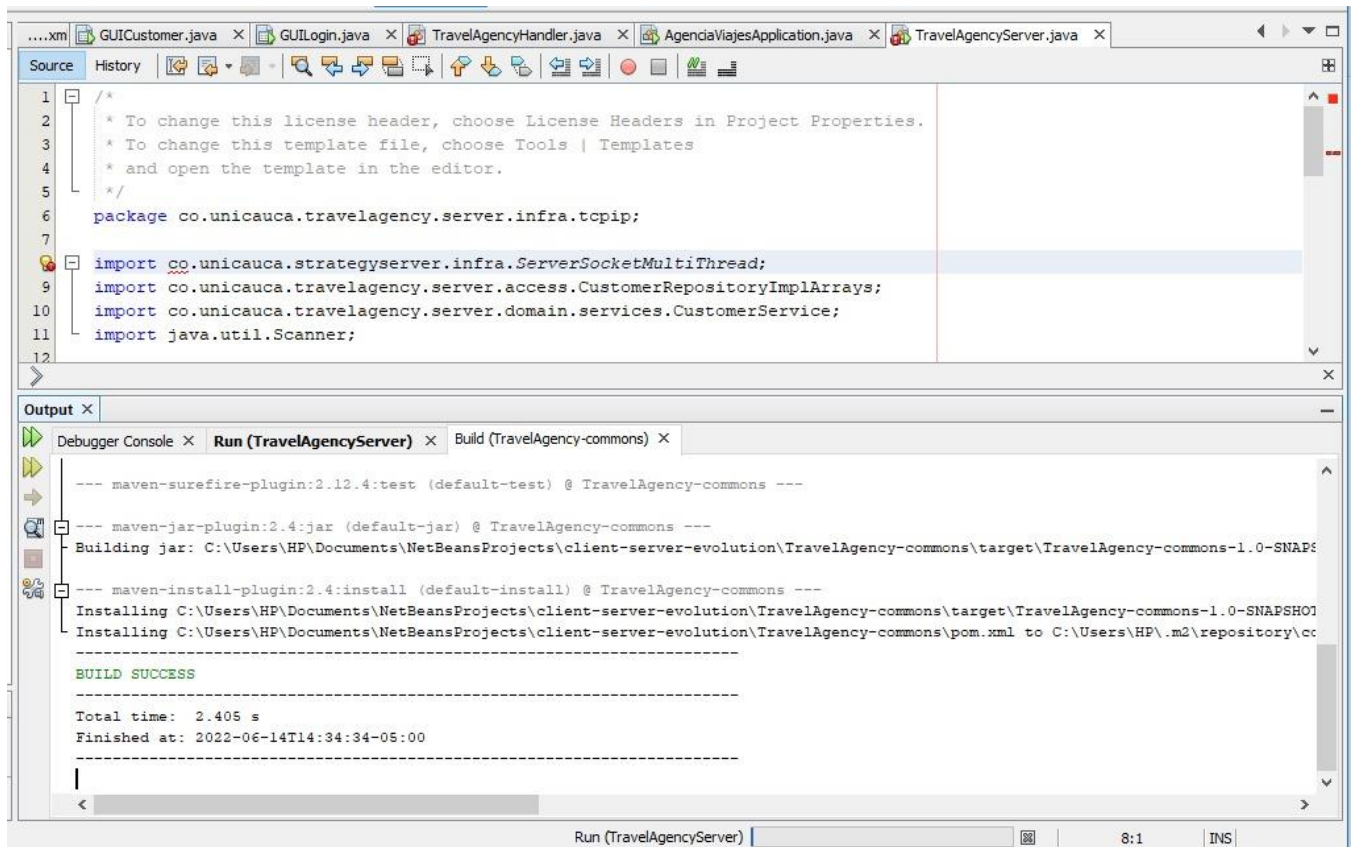
```
Progress (4): 164 kB | 323/335 kB | 57/61 kB | 128 kB
Progress (4): 164 kB | 327/335 kB | 57/61 kB | 128 kB
Progress (4): 164 kB | 331/335 kB | 57/61 kB | 128 kB
Progress (4): 164 kB | 335 kB | 57/61 kB | 128 kB

Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-artifact-transfer/0.10.1/maven-arti
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.1.0/maven-shared-uti
Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.11/commons-codec-1.11.jar (335 kB)
Progress (1): 60/61 kB
Progress (1): 61 kB

Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-common-artifact-filters/3.0.1/maver
jun. 14, 2022 1:17:13 P. M. co.unicauca.strategyserver.infra.ServerSocketMultiThread openPort
INFO: Servidor iniciado, escuchando por el puerto 5.000
En servidor multihilo esta espera
```

The status bar at the bottom of the IDE shows 'Run (EchoServerExample)', a magnifying glass icon, '1:1', and 'INS'.

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



Agencia de viajes

OpcionesInformesConfiguraciónAyuda (Invitado)

Consulta de Clientes

Este ejercicio aplica el patrón cliente/servidor.
La aplicación cliente se conecta al servidor mediante Sockets.
El servidor devuelve el objeto Cliente consultado en formato JSON.
En el backend las cédulas desde 98000001 hasta 98000010.

Número de identificación: 98000001

Buscar

*Nombres: Andrea

*Apellidos: Sanchez

Dirección: Calle 14 No 11-12 Popayan

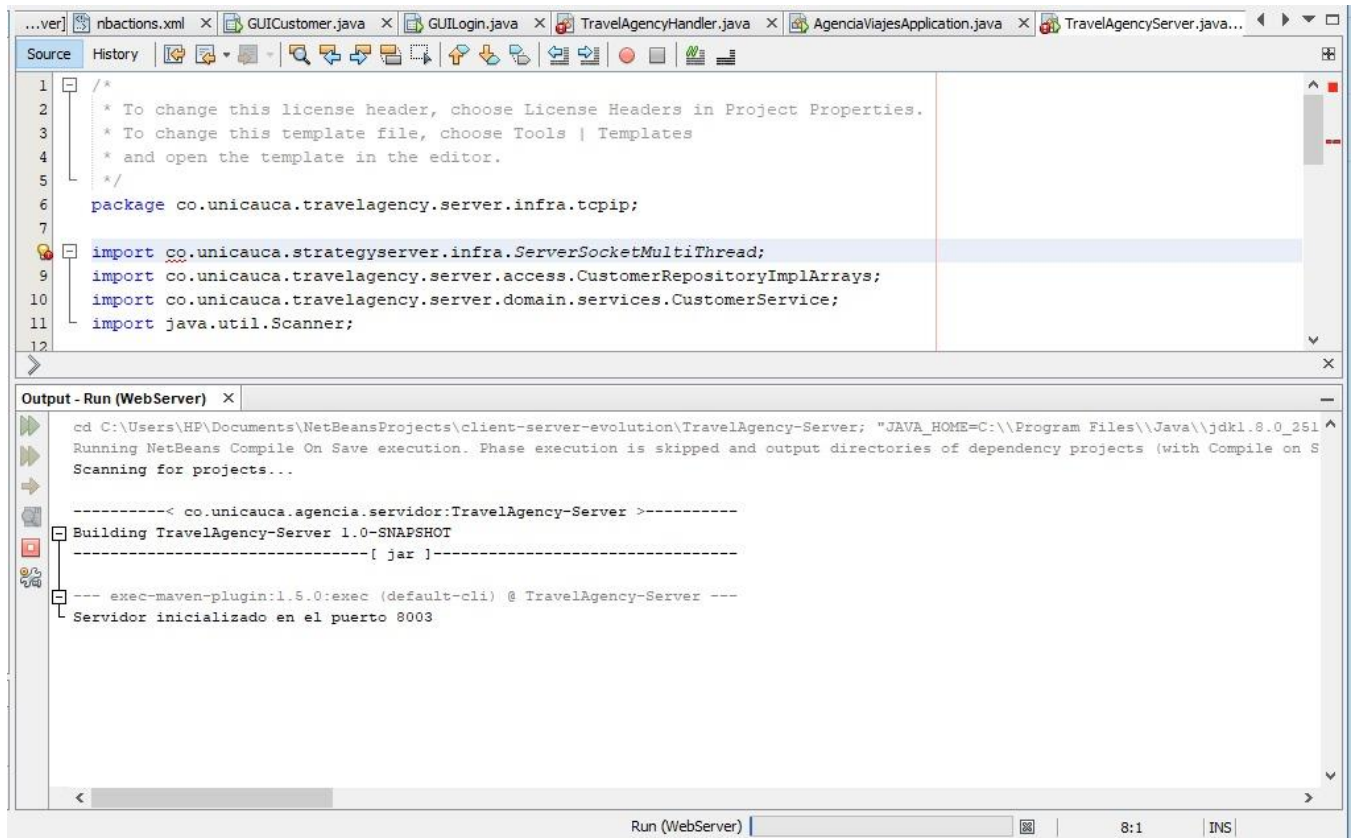
Celular: 3145878752

Email: andrea@hotmail.com

*Sexo (M, F): Femenino

Cerrar

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



GET http://localhost:8003/c

No Environment

http://localhost:8003/client

Save

</>

GET

http://localhost:8003/client

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

```
{ "parameters": [ { "name": "id", "value": "98000007" } ] }
```

Body

Cookies

Headers (2)

Test Results

Status: 200 OK

Time: 47 ms

Size: 252 B

Save Response

Pretty

Raw

Preview

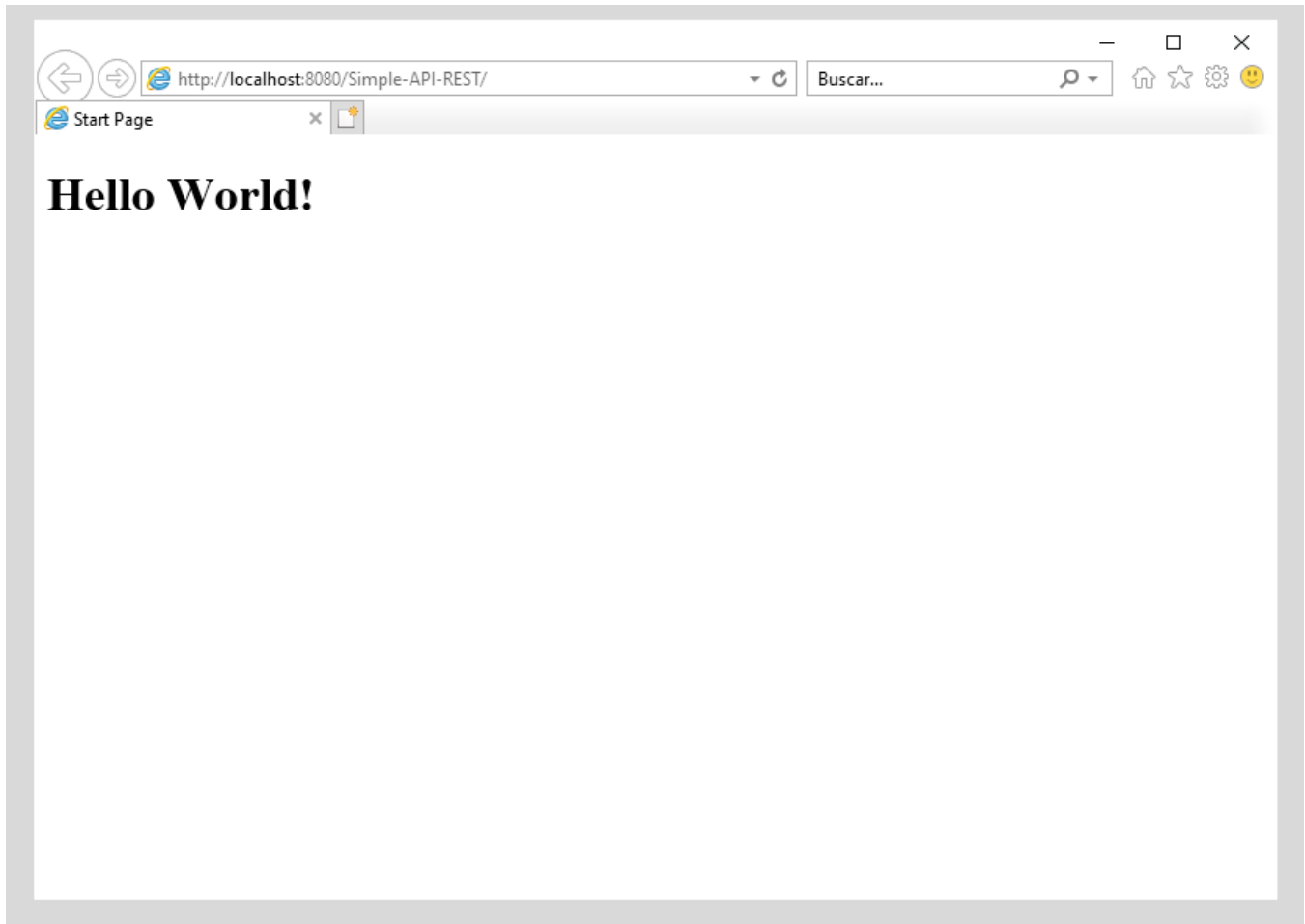
Visualize

Text

1

```
{ "id": "98000007", "firstName": "Cesar", "lastName": "Gutierrez Sanchez", "address": "Calle 12 No 12-12 Popayan", "mobile": "3154575845", "email": "fer@hotmail.com", "gender": "Masculino" }
```

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



- ✓ GET (consulta y lectura)
- Consultar todos los productos

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

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

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

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

This request does not have a body

Body Cookies Headers (5) Test Results Status: 200 OK Time: 2.01 s Size: 392 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "name": "Tv samsung",
5     "price": 200000.0
6   },
7   {
8     "id": 2,
9     "name": "Tv lg",
10    "price": 300000.0
11  },
12  {
13    "id": 1,
14    "name": "Tablet asus RESD-TD-34",
15    "price": 400000.0
16  }
```

-Consultar un producto en particular, en este ejemplo con id=1

GET http://localhost:8080/Simple-API-REST/product-service/products/1 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

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

This request does not have a body

Body Cookies Headers (5) Test Results Status: 200 OK Time: 68 ms Size: 290 B Save Response

Pretty Raw Preview Visualize JSON

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


✓ POST (crear)

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

- Method:** POST
- URL:** `http://localhost:8080/Simple-API-REST/product-service/products/`
- Body:** `{ "id": 99, "name": "Portatil Lenovo TinkPad", "price": 300000.0 }`
- Response:** `{ "ok": "true", "mensaje": "Producto creado", "errores": "" }`
- Status:** 200 OK
- Time:** 205 ms
- Size:** 294 B

✓ DELETE (eliminar)

-Eliminar un producto en particular, para este ejemplo con id=2

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

- Method:** DELETE
- URL:** `http://localhost:8080/Simple-API-REST/product-service/products/2`
- Body:** This request does not have a body
- Response:** `{ "ok": "true", "mensaje": "Producto borrado", "errores": "" }`
- Status:** 200 OK
- Time:** 257 ms
- Size:** 295 B

✓ PUT (editar).

-Para este caso se editó el producto creado anteriormente

The screenshot shows a Postman interface for a PUT request. The URL is `http://localhost:8080/Simple-API-REST/product-service/products/`. The request body is a JSON object: `{ "id": 99, "name": "Portatil Mac", "price": 600000.0 }`. The response status is 200 OK, and the response body is: `{ "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.

✓ POST (crear)

-En este caso creamos varios usuarios para luego listarlos

The screenshot shows a Postman interface for a POST request. The URL is `http://localhost:8080/TravelAgency-API-REST/customer-service/customers/`. The request body is a JSON object: `{ "address": "calle 13 No 7-87 Jamundi", "email": "abcde@unicauca.edu.co", "firstName": "Sebastian", "gender": "Masculino", "id": 90000123, "lastName": "Osorio", "mobile": "Seb3167498289" }`. The response status is 200 OK, and the response body is: `{ "ok": "true", "mensaje": "Cliente creado", "errores": "" }`.

POST

http://localhost:8080/TravelAgency-API-REST/customer-service/customers/

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

2

3

4

5

6

7

8

9

10

....

"address":

..

"carera 135 No 78-87 Popayán",

....

"email":

..

"fghi@unicauca.edu.co",

....

"firstName":

..

"Andes",

....

"gender":

..

"Masculino",

....

"id":

..

90000456,

....

"lastName":

..

"Zapata",

....

"mobile":

..

"3167568289"

....

10

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 69 ms

Size: 293 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

10

"ok":

"true",

"mensaje":

"Cliente creado",

"errores":

""

2

POST

http://localhost:8080/TravelAgency-API-REST/customer-service/customers/

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

2

3

4

5

6

7

8

9

10

....

"address":

..

"carera 5 No 7-7 Cali",

....

"email":

..

"jhkl@unicauca.edu.co",

....

"firstName":

..

"Camila",

....

"gender":

..

"Femenino",

....

"id":

..

90000789,

....

"lastName":

..

"Ortiz",

....

"mobile":

..

"3167568783"

....

10

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 51 ms

Size: 293 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

10

"ok":

"true",

"mensaje":

"Cliente creado",

"errores":

""

2

✓ GET (consultar)

-Consultamos a un usuario en específico, para este caso con id=90000789

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/TravelAgency-API-REST/customer-service/customers/90000789`. The request is sent, and the response is displayed in the 'Body' tab. The response is a JSON object with the following fields:

```
1 {
2   "address": "carera 5 No 7-7 Cali",
3   "email": "jhkl@unicauca.edu.co",
4   "firstName": "Camila",
5   "gender": "Femenino",
6   "id": 90000789,
7   "lastName": "Ortiz",
8   "mobile": "3167568783"
9 }
```

The status of the response is 200 OK, with a time of 40 ms and a size of 407 B. The response is saved.

✓ Repositorio del código:

<https://github.com/juanoa26/TravelAgency-API-REST>