



Universidad del Cauca

Taller

Presentado A:

Julio Ariel Hurtado

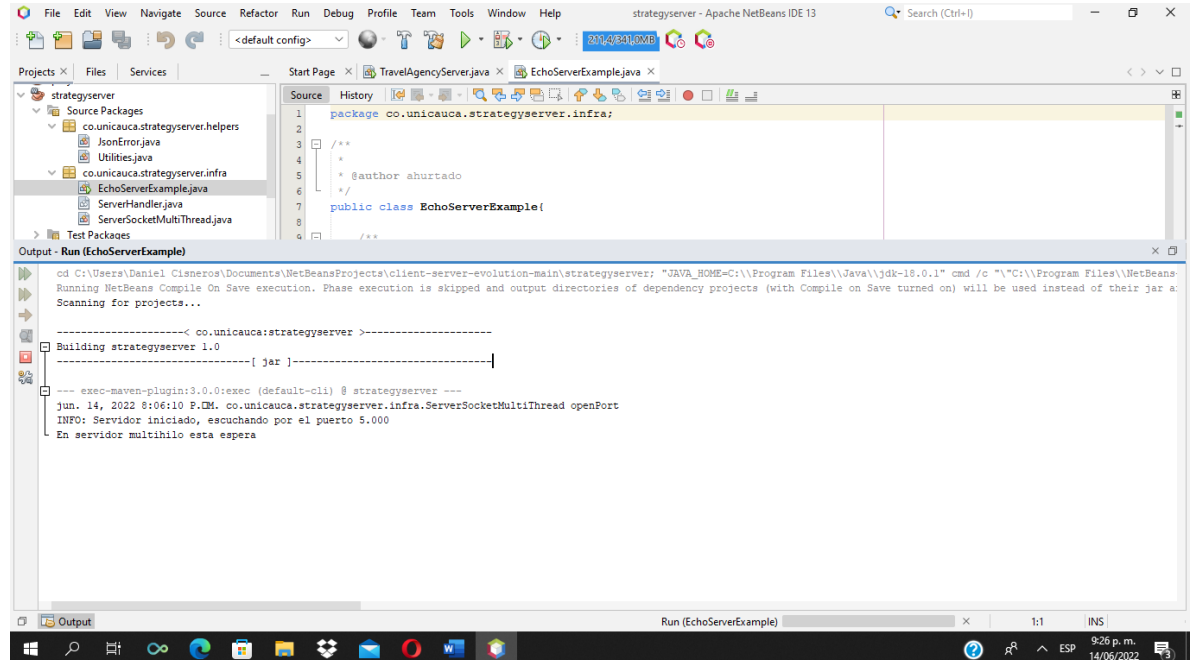
Presentado por:

Daniel Mauricio Cisneros Silva

Popayán, Cauca

22/06/2022

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



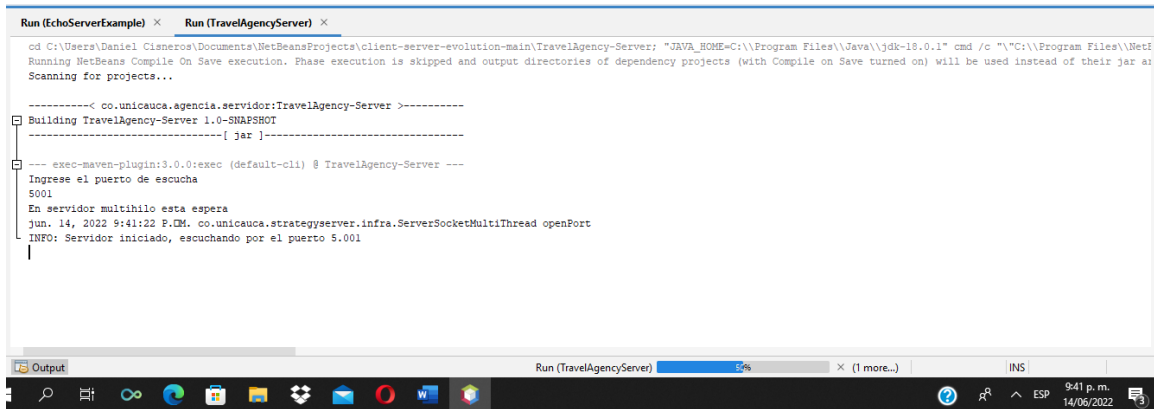
it - Run (EchoServerExample)

```
cd C:\Users\Daniel Cisneros\Documents\NetBeansProjects\client-server-evolution-main\strategyserver; "JAVA
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of depende
Scanning for projects...

-----< co.unicauca:strategyserver >-----
] Building strategyserver 1.0
-----[ jar ]-----

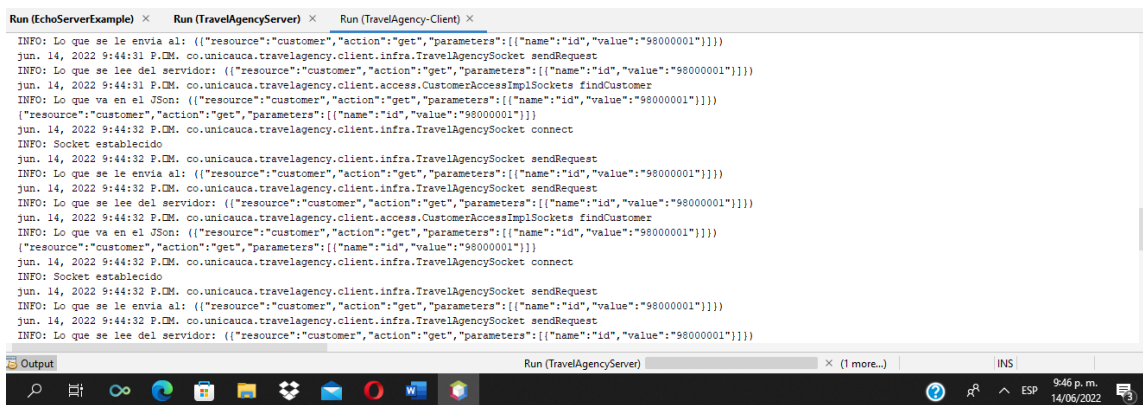
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ strategyserver ---
jun. 14, 2022 8:06:10 P.M. co.unicauca.strategyserver.infra.ServerSocketMultiThread openPort
INFO: Servidor iniciado, escuchando por el puerto 5.000
- En servidor multihilo esta espera
```

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



```
Run (EchoServerExample) × Run (TravelAgencyServer) ×
cd C:\Users\Daniel Cisneros\Documents\NetBeansProjects\client-server-evolution-main\TravelAgency-Server; "JAVA_HOME=C:\Program Files\Java\jdk-18.0.1" cmd /c "%C:\Program Files\Net\
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save turned on) will be used instead of their jar as
Scanning for projects...

-----< co.unicauca.agencia.servidor:TravelAgency-Server >-----
Building TravelAgency-Server 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ TravelAgency-Server ---
Ingresar el puerto de escucha
5001
En servidor multihilo esta espera
jun. 14, 2022 9:41:22 P.M. co.unicauca.strategyserver.infra.ServerSocketMultiThread openPort
INFO: Servidor iniciado, escuchando por el puerto 5.001
|
```



```
Run (EchoServerExample) × Run (TravelAgencyServer) × Run (TravelAgency-Client) ×
INFO: Lo que se le envia al: ({"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]})
jun. 14, 2022 9:44:31 P.M. co.unicauca.travelagency.client.infra.TravelAgencySocket sendRequest
INFO: Lo que se lee del servidor: ({"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]})
jun. 14, 2022 9:44:31 P.M. co.unicauca.travelagency.client.access.CustomerAccessImplSockets findCustomer
INFO: Lo que va en el JSON: ({"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]})
{"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]}
jun. 14, 2022 9:44:32 P.M. co.unicauca.travelagency.client.infra.TravelAgencySocket connect
INFO: Socket establecido
jun. 14, 2022 9:44:32 P.M. co.unicauca.travelagency.client.infra.TravelAgencySocket sendRequest
INFO: Lo que se le envia al: ({"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]})
jun. 14, 2022 9:44:32 P.M. co.unicauca.travelagency.client.infra.TravelAgencySocket sendRequest
INFO: Lo que se lee del servidor: ({"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]})
jun. 14, 2022 9:44:32 P.M. co.unicauca.travelagency.client.access.CustomerAccessImplSockets findCustomer
INFO: Lo que va en el JSON: ({"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]})
{"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]}
jun. 14, 2022 9:44:32 P.M. co.unicauca.travelagency.client.infra.TravelAgencySocket connect
INFO: Socket establecido
jun. 14, 2022 9:44:32 P.M. co.unicauca.travelagency.client.infra.TravelAgencySocket sendRequest
INFO: Lo que se le envia al: ({"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]})
jun. 14, 2022 9:44:32 P.M. co.unicauca.travelagency.client.infra.TravelAgencySocket sendRequest
INFO: Lo que se lee del servidor: ({"resource":"customer","action":"get","parameters":[{"name":"id","value":"98000001"}]})
```

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 cedulas desde 98000001 hasta 98000010.

Número de identificación:

*Nombres: Carlos

*Apellidos: Pantoja

Dirección: Santa Barbar Popayan

Celular: 3141257846

Email: carlos@gmail.com

*Sexo (M, F): Masculino

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. Ver la consulta postman.

Run (EchoServerExample) × Run (TravelAgencyServer) × Run (WebServer) ×

```

d C:\Users\Daniel Cisneros\Documents\NetBeansProjects\client-server-evolution-main\TravelAgency-Server: "JAVA_HOME=C:\Program Files\Java\jdk-18.0.1" cmd /c "%C:\Program Files\Net
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save turned on) will be used instead of their jar as
Scanning for projects...

-----< co.unicauca.agencia.servidor:TravelAgency-Server >-----
Building TravelAgency-Server 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ TravelAgency-Server ---
Servidor inicializado en el puerto 8003
{"parameters":[{"name":"id","value":"98000003"}]}

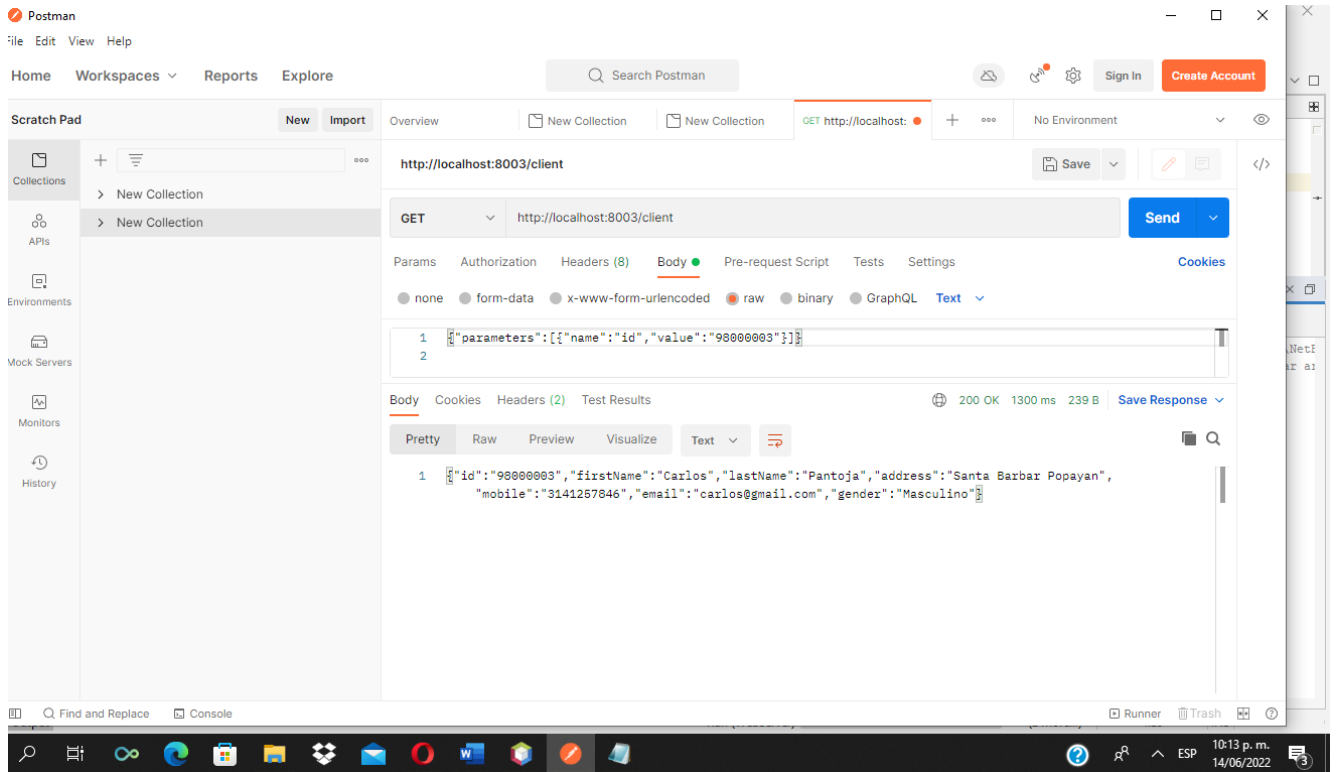
```

Output

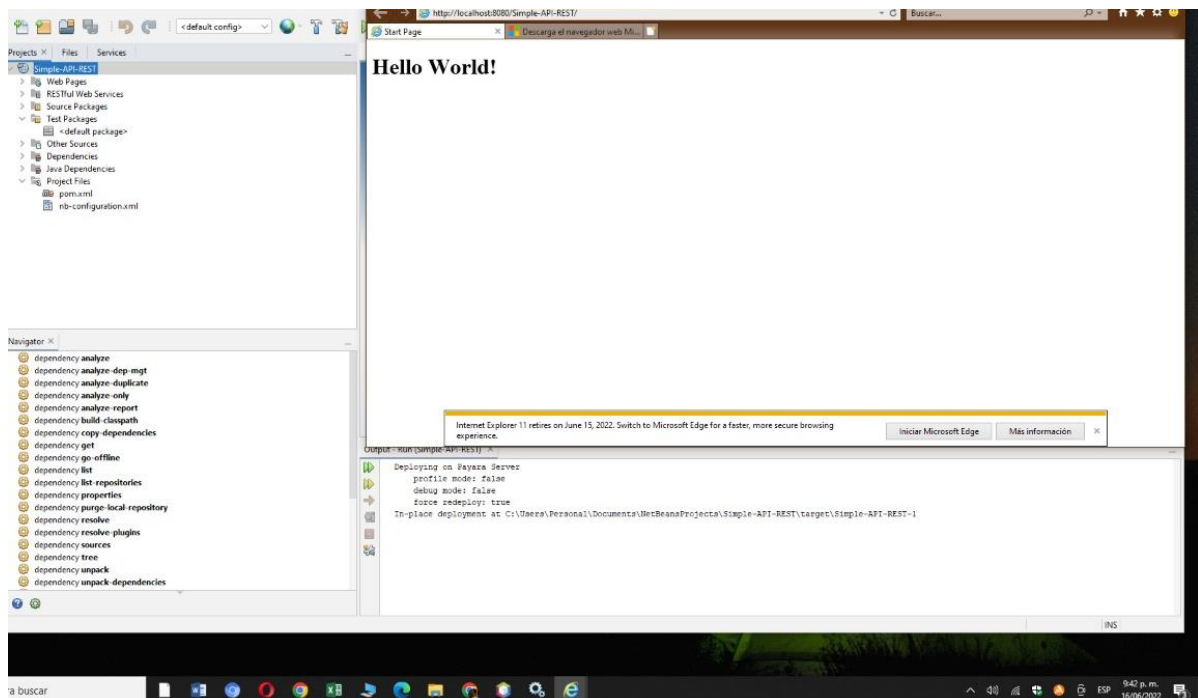
Run (WebServer) × (2 more...) 4:25 INS

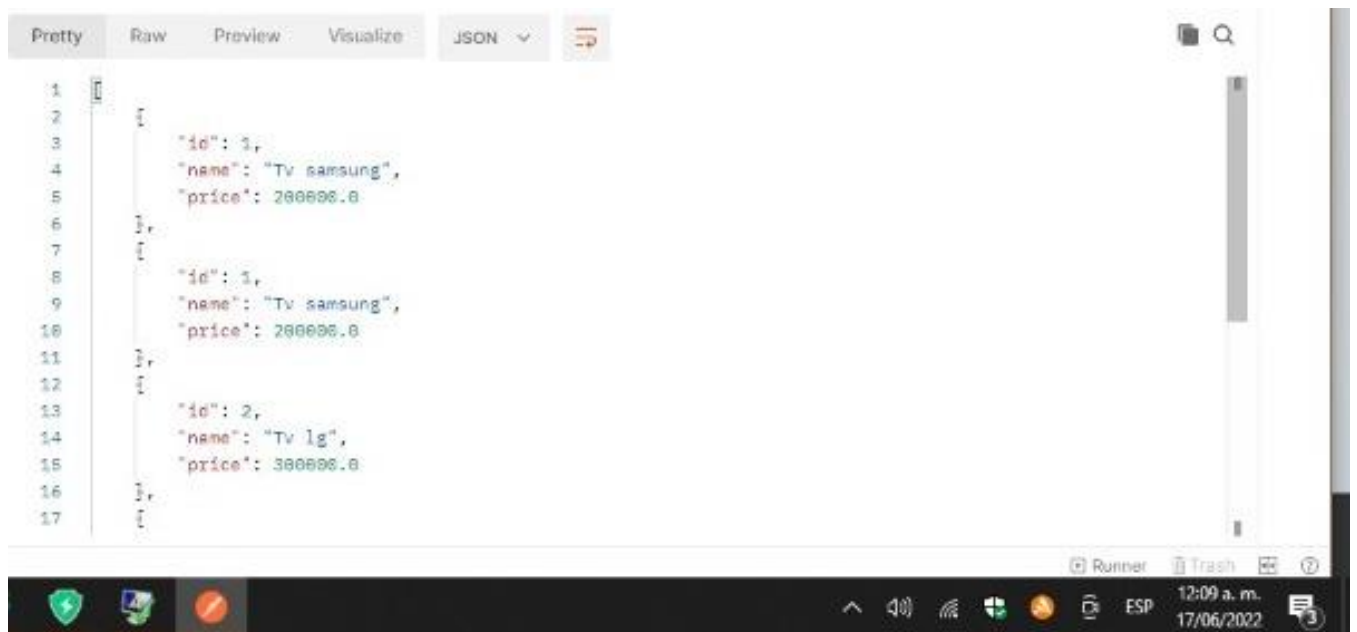
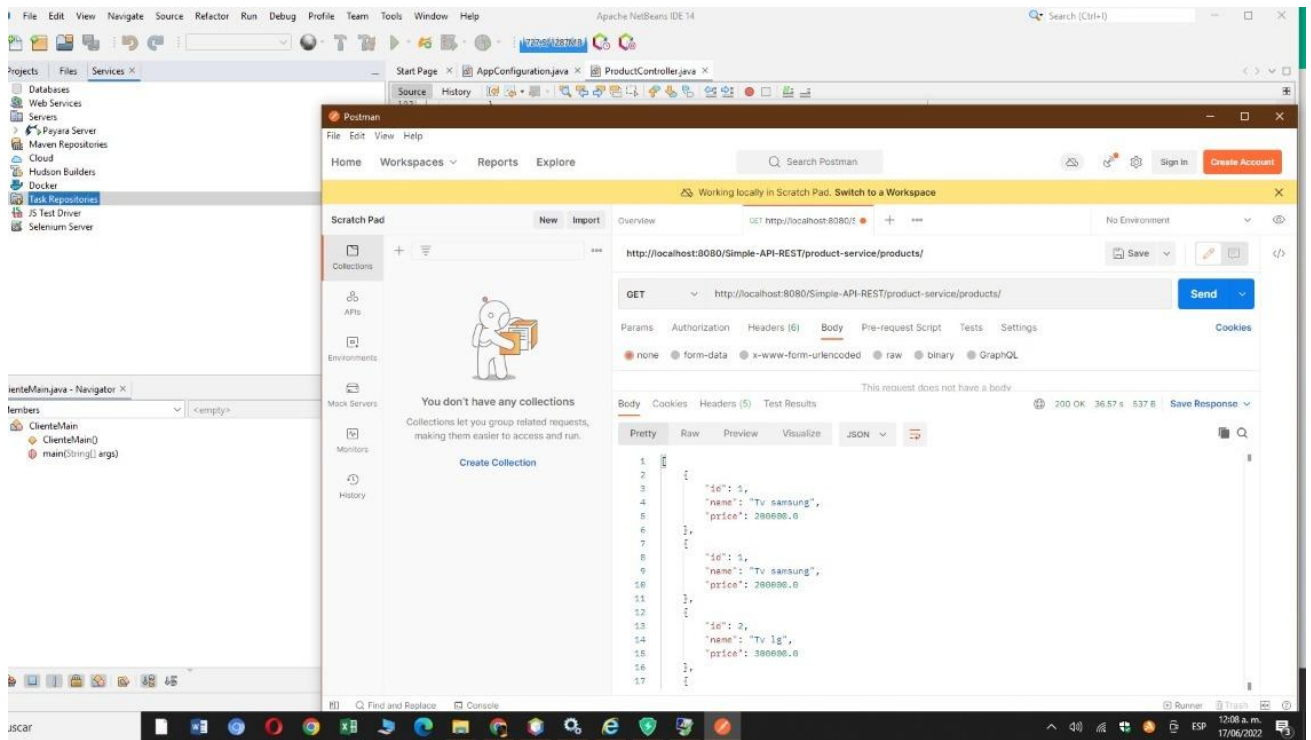
10:13 p. m.

14/06/2022



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





Buscar Producto con id= 2

GET ▼ http://localhost:8080/Simple-API-REST/product-service/products/2 Send ▼

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

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 107 ms 285 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 {
2   "id": 2,
3   "name": "Tv lg",
4   "price": 300000.0
5 }
```

Añadiendo un Producto

POST ▼ 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 {
2   "id": 6,
3   "name": "Xbox 360",
4   "price": 480.000
5 }
```

Body Cookies Headers (5) Test Results 200 OK 137 ms 294 B Save Response ▼

Pretty Raw Preview Visualize Text ▼ ≡

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

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

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

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

Body Cookies Headers (5) Test Results

200 OK 13 ms 433 B S

Pretty Raw Preview Visualize JSON

```
13  "id": 1,  
14    "name": "Tablet asus RESD-TD-34",  
15    "price": 400000.0  
16  },  
17  
18  "id": 6,  
19    "name": "Xbox 360",  
20    "price": 480.0  
21  ],  
22  ]
```

Ahora eliminamos el producto creado

DELETE http://localhost:8080/Simple-API-REST/product-service/products/6

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

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

Body Cookies Headers (5) Test Results

200 OK 39 ms 295 B S

Pretty Raw Preview Visualize Text

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


Buscamos el producto con id 6 para verificar si se borró correctamente

GET

⌵

http://localhost:8080/Simple-API-REST/product-service/products/6

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	D
	Key	Value	D

Body

Cookies

Headers (3)

Test Results

⌵

204 No C

Pretty

Raw

Preview

Visualize

Text ⌵

≡ ↺

1

Ahora modifiquemos el Producto con ID=1 el cual tiene la siguiente Información

GET

⌵

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

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

Query Params

	KEY	VALUE
	Key	Value

BodyCookiesHeaders (5)Test Results

PrettyRawPreviewVisualizeJSON⌵↺

```
1  [
2    {
3      "id": 1,
4      "name": "Tv Samsung",
5      "price": 500.0
6    },
```

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

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

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

```
1 {
2   ... "id": 1,
3   ... "name": "Equipo de sonido",
4   ... "price": 100.0
5 }
```

Body Cookies Headers (5) Test Results

200 OK 16 ms

Pretty Raw Preview Visualize Text

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

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

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

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

Body Cookies Headers (5) Test Results

200 OK 9 ms 373 B

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "name": "Equipo de sonido",
5     "price": 100.0
6   },
7 }
```

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

Link GitHub:

<https://github.com/danielcis25/Travel-Agency-API-Rest>

listar el cliente con ID 1

http://localhost:8080/TravelAgency-API-REST/customer-service/customers/1

GET http://localhost:8080/TravelAgency-API-REST/customer-service/customers/1

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

Query Params


KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results 200 OK 591 m

Pretty Raw Preview Visualize JSON

```
1 {
2   "address": "Carrera 14 # 1-12 Popayan",
3   "email": "andresC@hotmail.com",
4   "firstName": "Andres",
5   "gender": "Masculino",
6   "id": "1",
7   "lastName": "Castro",
8   "mobile": "3192258752"
9 }
```


Creamos un cliente

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

Params

Authorization


Headers (8)


Body 


Pre-request Script


Tests


Settings


 none


 form-data

 x-www-form-urlencoded

 **raw**

 binary

 GraphQL

JSON 


```
2  {
3    "address": "Carrera 50 #1-14 Popayan",
4    "email": "DanielC@hotmail.com",
5    "firstName": "Daniel",
6    "gender": "Masculino",
7    "id": "6",
8    "lastName": "C",
9    "mobile": "3192445890"
10 }
```

Body

Cookies

Headers (5)

Test Results


 200


Pretty

Raw


Preview

Visualize

Text 



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

GET  http://localhost:8080/TravelAgency-API-REST/customer-service/customers/6


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

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

Body Cookies Headers (5) Test Results

 200 OK 30 ms 402

Pretty Raw Preview Visualize JSON  

```
1  {
2    "address": "Carrera 50 #1-14 Popayan",
3    "email": "DanielC@hotmail.com",
4    "firstName": "Daniel",
5    "gender": "Masculino",
6    "id": "6",
7    "lastName": "C",
8    "mobile": "3192445890"
9  }
```