

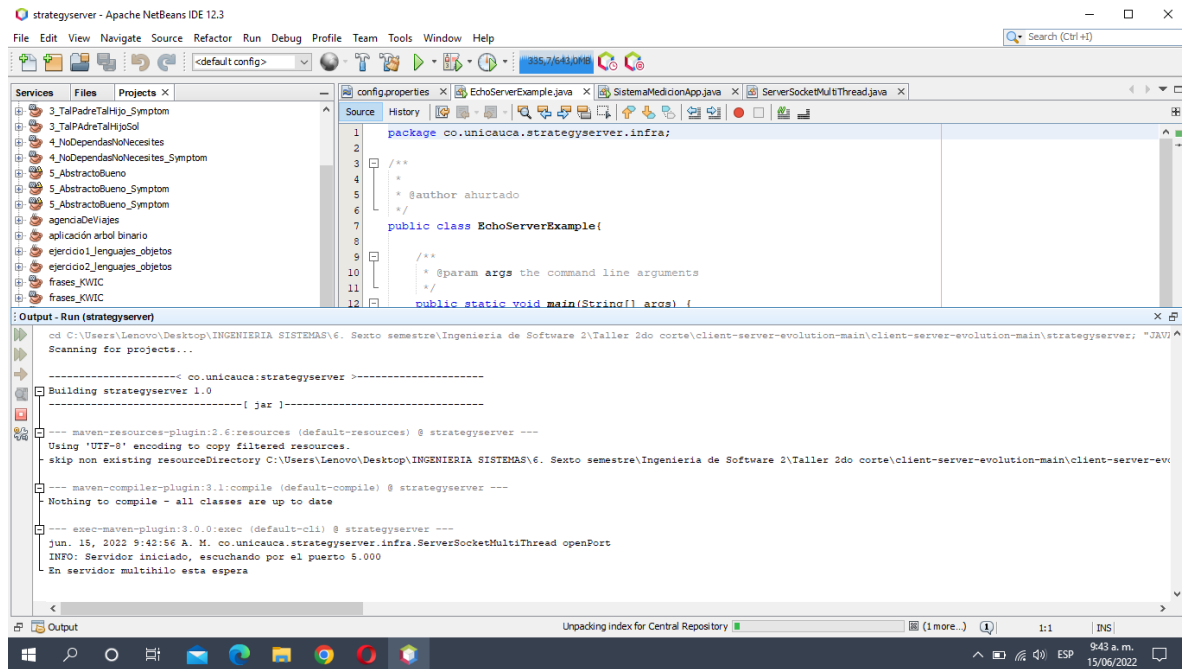
## Taller Cliente Servidor

Estudiante: Yazmin Viviana Gutierrez Sarria

Código: 104619011189

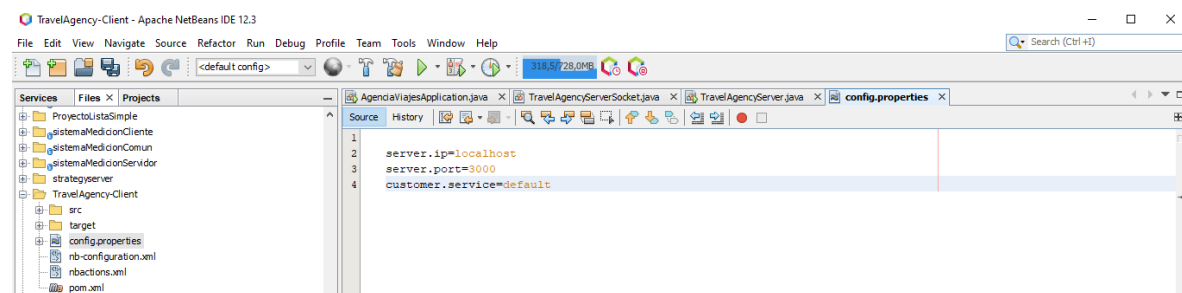
1. Descargar, instalar y correr la clase EchoServerExample que viene en el proyecto strategyserver.

Al ejecutar la clase EchoServerExample se puede observar que el servidor es iniciado en el puerto 5000.

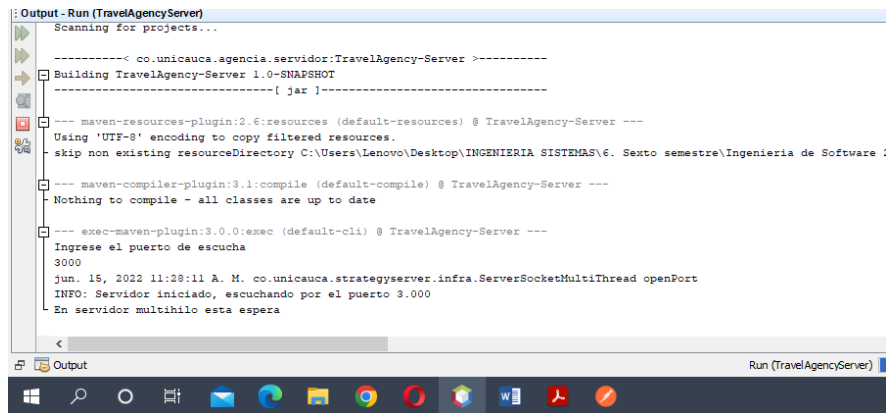


2. Descargar, instalar y correr el AgencyTravelServer como servidor tcp/ip y el AgencyTravel-Client (infra.tcpip).

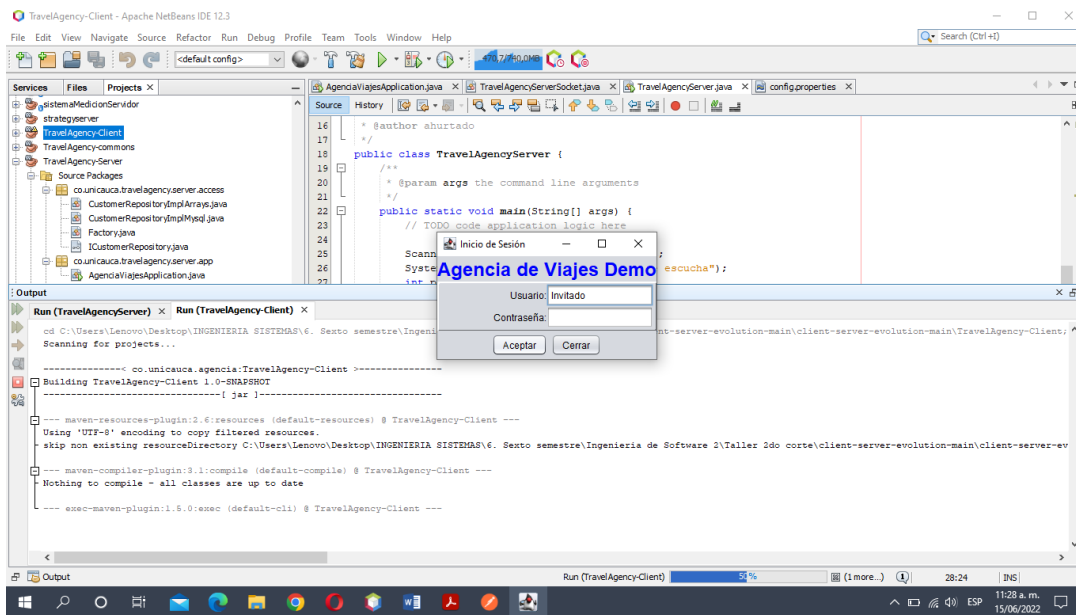
Para realizar este proceso se debe cambiar la configuración del puerto en el proyecto de cliente de tal forma que coincida con el puerto que se le va a asignar al servidor, esto se hace en el archivo `config.properties`.



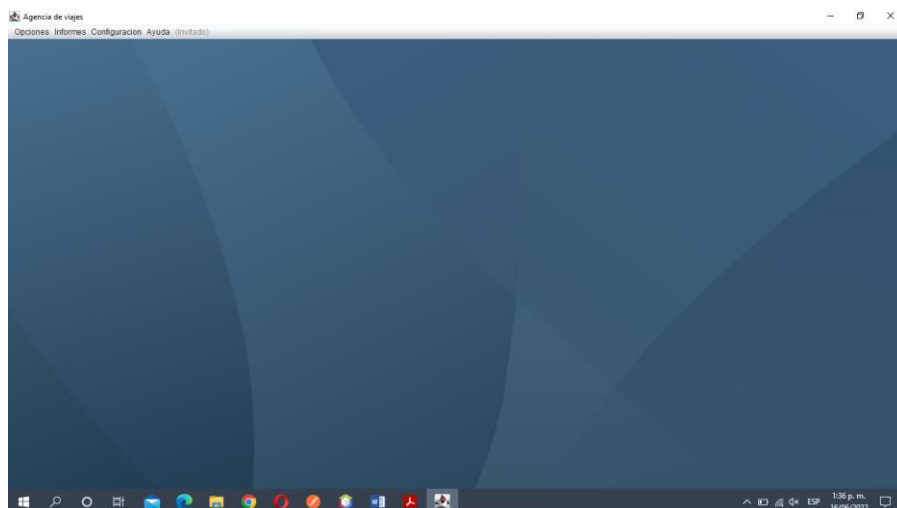
Una vez realizado el paso anterior se ejecuta el `TravelAgencyServer` para iniciar el servidor.



Posteriormente se ejecuta TravelAgencyClient y es aquí donde se puede visualizar la interfaz gráfica de inicio de Sesión a la cual se le da clic en aceptar.



Luego aparece una nueva interfaz la cual contiene el menú de opciones.

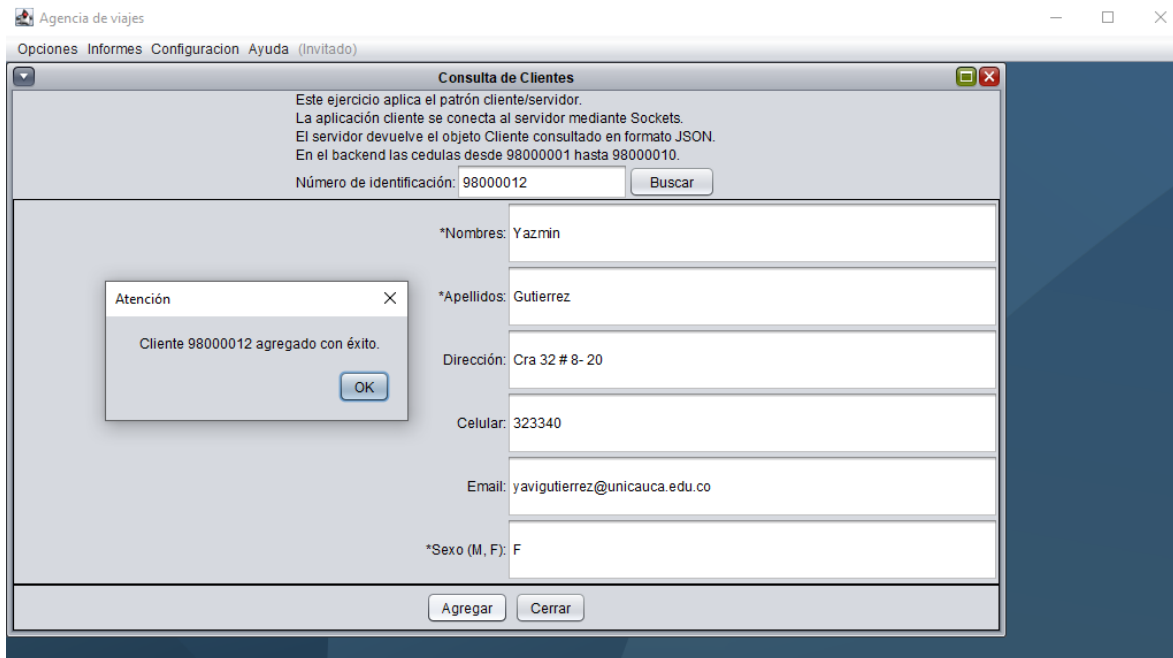


Al seleccionar **Opciones/Gestión** de clientes se puede acceder a la información de los clientes mediante su número de identificación como se muestra a continuación.

The screenshot shows a window titled 'Consulta de Clientes' from the 'Agencia de viajes' application. The window has a menu bar with 'Opciones', 'Informes', 'Configuracion', and 'Ayuda (Invitado)'. Inside, there is a text area with the following information: '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.' Below this, there is a search field with the text 'Número de identificación: 98000001' and a 'Buscar' button. The search results are displayed in a form with the following fields: '\*Nombres: Andrea', '\*Apellidos: Sanchez', 'Dirección: Calle 14 No 11-12 Popayan', 'Celular: 3145878752', 'Email: andrea@hotmail.com', and '\*Sexo (M, F): Femenino'. At the bottom of the form is a 'Cerrar' button.

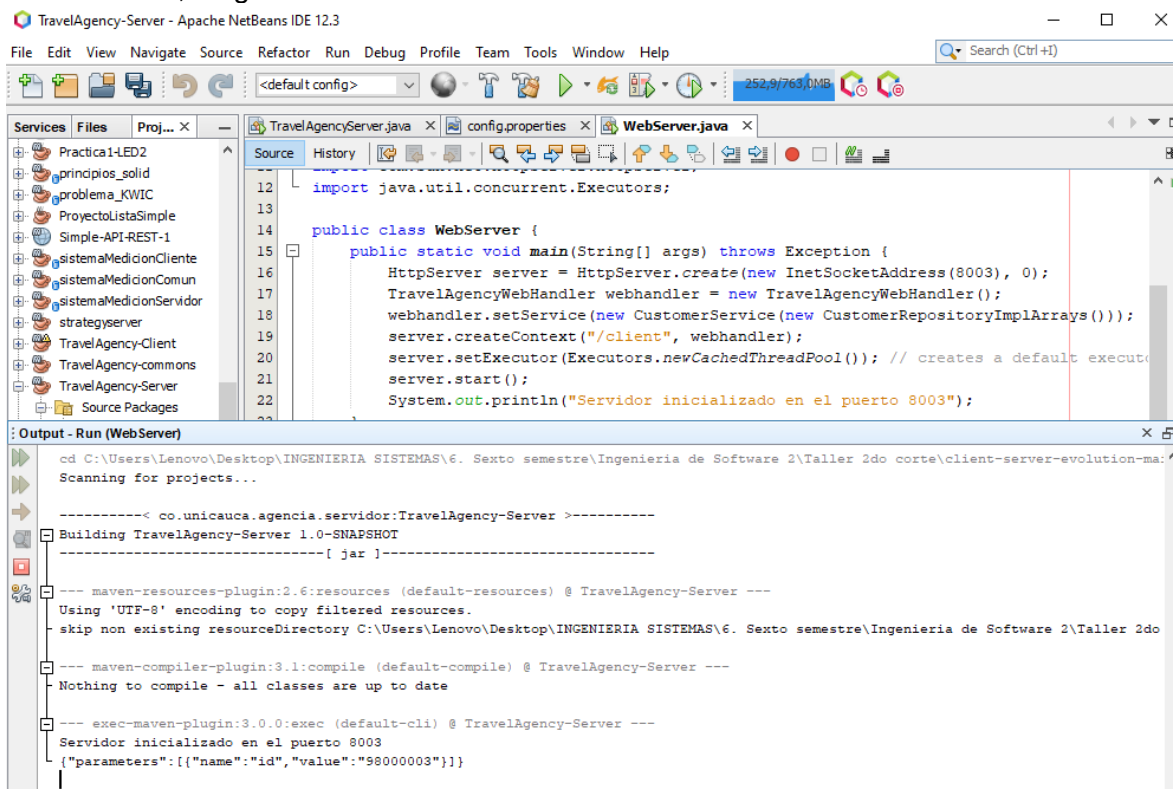
Se pueden agregar nuevos clientes si estos no se encuentran registrados como se muestra a continuación.

The screenshot shows the same 'Consulta de Clientes' window, but with the search field containing 'Número de identificación: 98000012'. The search results form is empty. An 'Atención' dialog box is open in the foreground, displaying the message 'Cliente no encontrado. Cédula no existe' and an 'OK' button. The background form fields are: '\*Nombres:', '\*Apellidos:', 'Dirección:', 'Celular:', 'Email:', and '\*Sexo (M, F):'. The 'Cerrar' button is at the bottom.

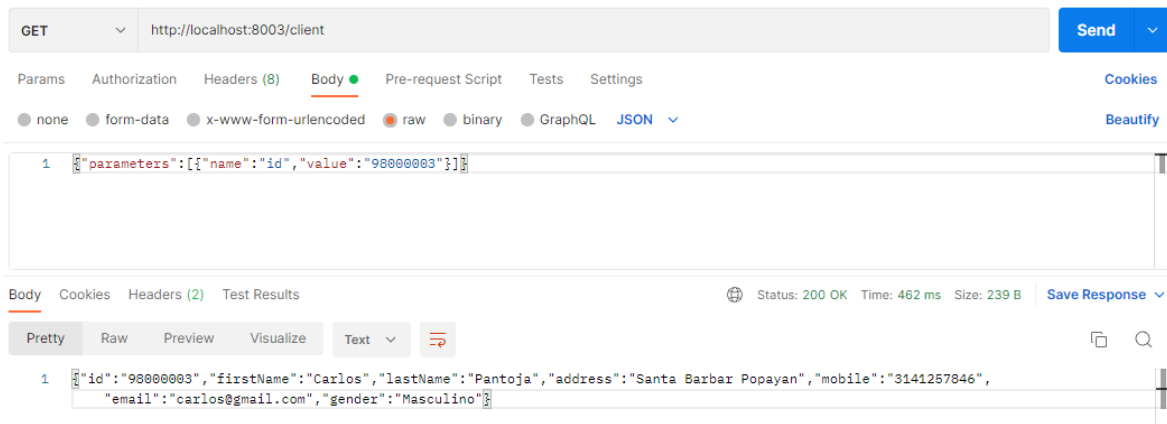


3. Correr el Web Server(infra.web) que vienen en el mismo proyecto AgencyTravelServer. Usar un cliente Postman para hacer la consulta, recuerden que el protocolo cambio y ahora solo van los parámetros. Ver la consulta Postman.

Para realizar este punto se descargó la herramienta **Postman** desde la página oficial, luego se corrió el **WebServer** como se muestra a continuación.

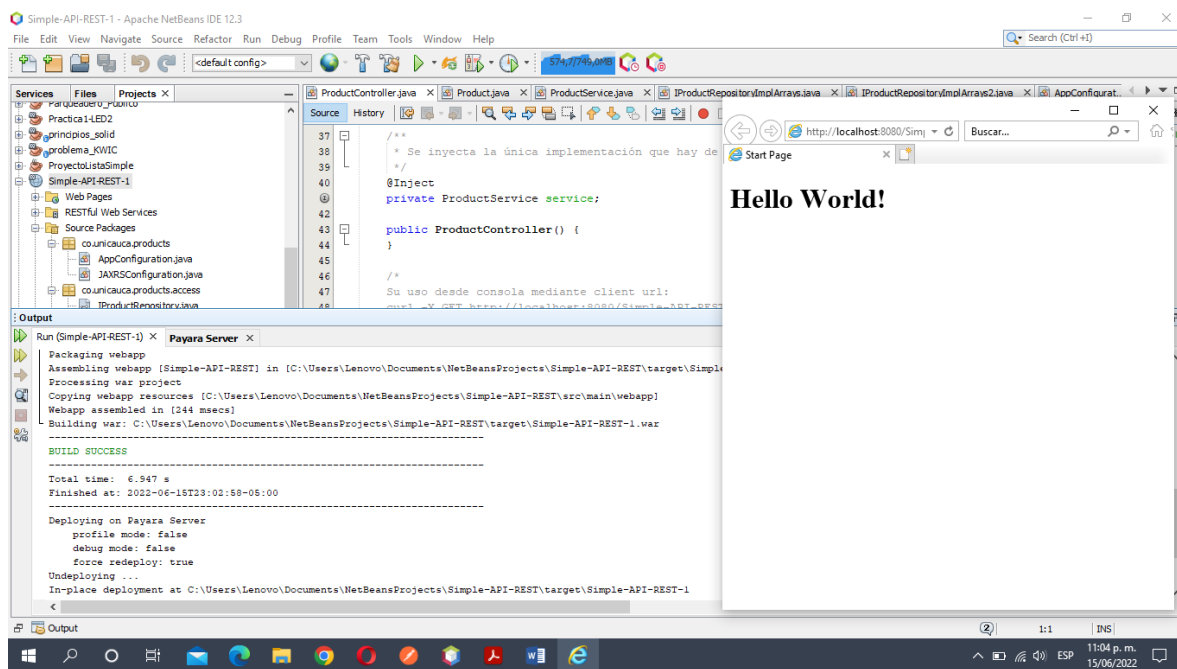


Finalmente se usó Postman para realizar la consulta de la información del cliente con id 9800003. Para procesar la petición del cliente se deben pasar los parámetros como se muestra a continuación.



4. Realizar el taller de la API Rest y probar las consultas a través de Postman o un Jersey Client. Sigán el paso a paso del taller.

Antes de realizar las consultas por medio del Postman o del JerseyClient es necesario correr la Web Application “Simple-API-REST” para que se inicialice el servidor de aplicaciones Payara como se puede observar una vez aparece el mensaje “Hello World!” en el navegador.



## Probando las consultas a través de Postman:

- Método GET para listar todos los productos.

GET ▼ http://localhost:8080/Simple-API-REST/product-service/products/ 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 🌐 Status: 200 OK Time: 422 ms Size: 386 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ ≡

```
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 }
```

- Metodo GET para listar un producto en particular, en este caso 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

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description |     |           |

Body Cookies Headers (5) Test Results 🌐 Status: 200 OK Time: 25 ms Size: 284 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ ≡

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

- Método POST para crear 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 [{"id":3,"name":"Nevera Lg","price":6700000}]
```

Body Cookies Headers (5) Test Results 🌐 Status: 200 OK Time: 22 ms Size: 288 B Save Response ▼

Pretty Raw Preview Visualize Text ▼ ≡

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

- Método DELETE para eliminar in producto con id=2

DELETE ▼ 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 🌐 Status: 200 OK Time: 23 ms Size: 289 B Save Response ▼

Pretty Raw Preview Visualize Text ▼ ≡

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

Al usar el metodo GET para listar productos, ya no se encuentra el producto con id=2.

Body Cookies Headers (5) Test Results 🌐 Status: 200 OK Time: 18 ms Size: 345 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": 3,
9     "name": "Tablet asus RESD-TD-34",
10    "price": 400000.0
11  }
12 ]
```

- Método PUT para editar el producto con id=3 creado anteriormente.

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":3,"name":"Nevera Lg REF. JDK3-34-343","price":2450000}]
```

Body Cookies Headers (5) Test Results 🌐 Status: 200 OK Time: 21 ms Size: 292 B Save Response ▼

Pretty Raw Preview Visualize Text ▼ ≡

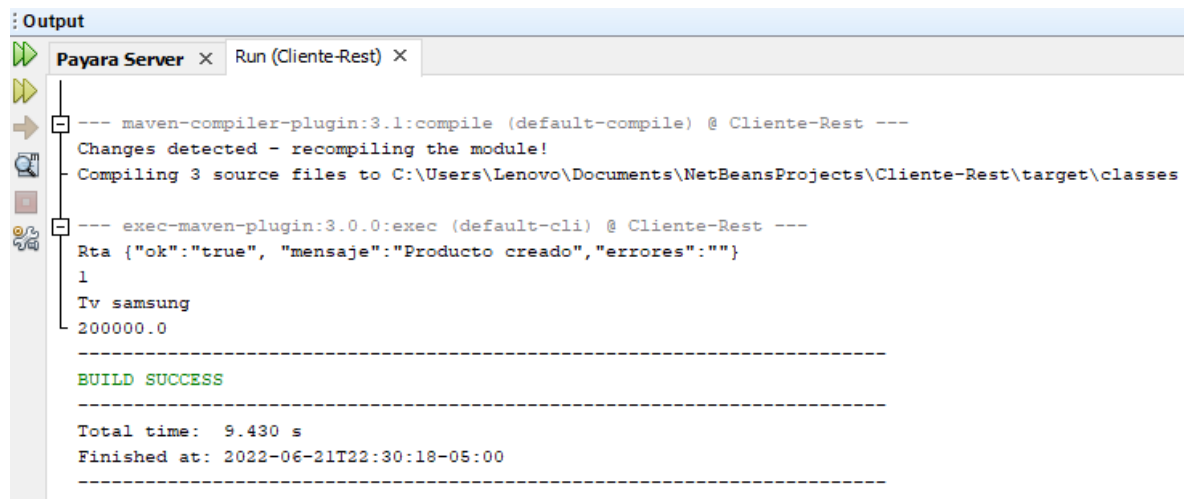
```
1 {"ok":"true", "mensaje":"Producto modificado","errores":""}
```

## Probando las consultas a través del Jersey Client:

Se crea un objeto de tipo `NewJerseyClient` para poder acceder a todos los métodos.

```
public class ClienteMain {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        //CREANDO UN PRODUCTO  
        NewJerseyClient jersey = new NewJerseyClient();  
        String rta = jersey.create_JSON(new Product(4, "Lavadora Samsung", 4570000d));  
        System.out.println("Rta " + rta);  
  
        //BUSCANDO UN PRODUCTO  
        Product product = jersey.findById(Product.class, "1");  
        System.out.println(product.getId());  
        System.out.println(product.getName());  
        System.out.println(product.getPrice());  
  
        //ELIMINAR UN PRODUCTO  
        Product product2 = jersey.findById(Product.class, "2");  
        String respuesta = jersey.remove(product2.getId().toString());  
        System.out.println("Rta " + respuesta);  
  
        //EDITAR UN PRODUCTO  
        Product product3 = jersey.findById(Product.class, "1");  
        product3.setName("Tv Caixun");  
        product3.setPrice(2500000d);  
        String res = jersey.update_JSON(product3);  
        System.out.println("Rta " + res);  
    }  
}
```

Salida por consola al ejecutar los métodos de crear y buscar producto.



```
Output  
Payara Server x Run (Cliente-Rest) x  
--- maven-compiler-plugin:3.1:compile (default-compile) @ Cliente-Rest ---  
Changes detected - recompiling the module!  
Compiling 3 source files to C:\Users\Lenovo\Documents\NetBeansProjects\Cliente-Rest\target\classes  
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Cliente-Rest ---  
Rta {"ok":"true", "mensaje":"Producto creado","errores":""}  
1  
Tv samsung  
200000.0  
-----  
BUILD SUCCESS  
-----  
Total time: 9.430 s  
Finished at: 2022-06-21T22:30:18-05:00  
-----
```



Salida por consola al ejecutar el método de eliminar producto.

```
Output
Payara Server x Run (Cliente-Rest) x
--- maven-resources-plugin:2.6:resources (default-resources) @ Cliente-Rest ---
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory C:\Users\Lenovo\Documents\NetBeansProjects\Cliente-Rest\src\mai
--- maven-compiler-plugin:3.1:compile (default-compile) @ Cliente-Rest ---
Changes detected - recompiling the module!
Compiling 3 source files to C:\Users\Lenovo\Documents\NetBeansProjects\Cliente-Rest\target\classes
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Cliente-Rest ---
Rta {"ok":"true", "mensaje":"Producto borrado","errores":""}

BUILD SUCCESS

Total time: 8.954 s
Finished at: 2022-06-21T22:39:07-05:00
```

Salida por consola al ejecutar el método de editar producto.

```
Output
Payara Server x Run (Cliente-Rest) x
--- maven-resources-plugin:2.6:resources (default-resources) @ Cliente-Rest ---
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory C:\Users\Lenovo\Documents\NetBeansProjects\Cliente-Rest\src\mai
--- maven-compiler-plugin:3.1:compile (default-compile) @ Cliente-Rest ---
Changes detected - recompiling the module!
Compiling 3 source files to C:\Users\Lenovo\Documents\NetBeansProjects\Cliente-Rest\target\classes
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Cliente-Rest ---
Rta {"ok":"true", "mensaje":"Producto modificado","errores":""}

BUILD SUCCESS

Total time: 9.073 s
Finished at: 2022-06-21T22:40:52-05:00
```

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

A continuación, se prueban todos los métodos desde el Postman:

- Método GET para listar todos los clientes.

GET http://localhost:8080/Simple-API-REST-Travel-Agency/product-service/services/ Send

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

Query Params

| KEY | VALUE | DESCRIPTION | *** | Bulk Edit |
|-----|-------|-------------|-----|-----------|
|-----|-------|-------------|-----|-----------|

Body Cookies Headers (5) Test Results Status: 200 OK Time: 416 ms Size: 1.89 KB Save Response

Pretty Raw Preview Visualize

```
[{"address":"Calle 14 No 11-12 Popayan","email":"andrea@hotmail.com","firstName":"Andrea","gender":"Femenino","id":"98000001","lastName":"Sanchez","mobile":"3145878752"}, {"address":"Santa Barbar Popayan","email":"libardo@gmail.com","firstName":"Libardo","gender":"Masculino","id":"98000002","lastName":"Pantoja","mobile":"3141257845"}, {"address":"Santa Barbar Popayan","email":"carlos@gmail.com","firstName":"Carlos","gender":"Masculino","id":"98000003","lastName":"Pantoja","mobile":"3141257846"}, {"address":"Calle 16 No 12-12 Popayan","email":"fercha@hotmail.com","firstName":"Fernanda","gender":"Femenino","id":"98000004","lastName":"Arevalo","mobile":"3154562133"}, {"address":"Calle 12 No 12-12 Popayan","email":"fez@hotmail.com","firstName":"Manuel","gender":"Masculino","id":"98000005","lastName":"Perez","mobile":"3154575845"}, {"address":"Calle 12 No 12-12 Popayan","email":"fez@hotmail.com","firstName":"Alejandro","gender":"Masculino","id":"98000006","lastName":"Mosquera","mobile":"3154575845"}, {"address":"Calle 12 No 12-12 Popayan","email":"fez@hotmail.com","firstName":"Cesar","gender":"Masculino","id":"98000007","lastName":"Gutierrez Sanchez","mobile":"3154575845"}, {"address":"Calle 12 No 12-12 Popayan","email":"fez@hotmail.com","firstName":"Julio","gender":"Masculino","id":"98000008","lastName":"Bravo Bravo","mobile":"3154575845"}, {"address":"Calle 12 No 12-12 Popayan","email":"fez@hotmail.com","firstName":"Alberto","gender":"Masculino","id":"98000009","lastName":"Mendez Bravo","mobile":"3154575845"}, {"address":"Calle 12 No 12-12 Popayan","email":"fez@hotmail.com","firstName":"Alexander","gender":"Masculino","id":"98000010","lastName":"Ponce Yepes","mobile":"3154575845"}]
```

- Método GET para buscar un cliente con id 98000001.

GET ▼ <http://localhost:8080/Simple-API-REST-Travel-Agency/product-service/services/98000001> Send ▼

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

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
|-----|-------|-------------|-----|-----------|

Body Cookies Headers (5) Test Results 200 OK 24 ms 408 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍

```

1 {
2   "address": "Calle 14 No 11-12 Popayan",
3   "email": "andrea@hotmail.com",
4   "firstName": "Andrea",
5   "gender": "Femenino",
6   "id": "98000001",
7   "lastName": "Sanchez",
8   "mobile": "3145878752"
9 }
```

- Método POST para crear un cliente.

POST ▼ <http://localhost:8080/Simple-API-REST-Travel-Agency/product-service/services/> 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": "98000011", "firstName": "Yazmin", "lastName": "Gutierrez", "address": "Calle 32 No 8-20 Popayan",
  "mobile": "3233408378", "email": "yazmin@gmail.com", "gender": "Femenino" }
```

Body Cookies Headers (5) Test Results 200 OK 59 ms 287 B Save Response ▼

Pretty Raw Preview Visualize Text ▼ 🔍

```

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

- Método PUT para modificar el cliente con id=98000006.

PUT ▼ <http://localhost:8080/Simple-API-REST-Travel-Agency/product-service/services/> 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": "98000006",
3   "firstName": "Viviana",
4   "lastName": "Sarria",
5   "address": "Calle 32 No 8-20 Popayan",
6   "mobile": "3217296166",
7   "email": "yazmin@gmail.com",
8   "gender": "Femenino"
9 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 17 ms Size: 291 B Save Response ▼

Pretty Raw Preview Visualize Text ▼ 🔍

```

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

- Método DELETE para eliminar el cliente con id=98000002.

DELETE <http://localhost:8080/Simple-API-REST-Travel-Agency/product-service/services/98000002> 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 Status: 200 OK Time: 37 ms Size: 288 B Save Response

Pretty Raw Preview Visualize Text

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

A continuación, se pueden observar todas las modificaciones que surgieron a partir de todas las consultas realizadas anteriormente.

GET <http://localhost:8080/Simple-API-REST-Travel-Agency/product-service/services/...> Send

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

Query Params

| KEY | VALUE | *** | Bulk Edit |
|-----|-------|-----|-----------|
|     |       |     |           |

Body Cookies Headers (5) Test Results Save Response

Pretty Raw Preview Visualize

Cliente modificado con el PUT

Cliente eliminado con el DELETE = 98000002, por eso no se encuentra, pasa directamente al 98000003

```
[{"address": "Calle 14 No 11-12 Popayan", "email": "andrea@hotmail.com", "firstName": "Andrea", "gender": "Femenino", "id": "98000001", "lastName": "Sanchez", "mobile": "3145878752"}, {"address": "Santa Barbar Popayan", "email": "carlos@gmail.com", "firstName": "Carlos", "gender": "Masculino", "id": "98000003", "lastName": "Pantoja", "mobile": "3141257846"}, {"address": "Calle 16 No 12-12 Popayan", "email": "fercha@hotmail.com", "firstName": "Fernanda", "gender": "Femenino", "id": "98000004", "lastName": "Arevalo", "mobile": "3154562133"}, {"address": "Calle 12 No 12-12 Popayan", "email": "fer@hotmail.com", "firstName": "Manuel", "gender": "Masculino", "id": "98000005", "lastName": "Perez", "mobile": "3154575845"}, {"address": "Calle 32 No 8-20 Popayan", "email": "yazmin@gmail.com", "firstName": "Viviana", "gender": "Femenino", "id": "98000006", "lastName": "Sarria", "mobile": "3217296166"}, {"address": "Calle 12 No 12-12 Popayan", "email": "fer@hotmail.com", "firstName": "Cesar", "gender": "Masculino", "id": "98000007", "lastName": "Gutierrez Sanchez", "mobile": "3154575845"}, {"address": "Calle 12 No 12-12 Popayan", "email": "fer@hotmail.com", "firstName": "Julio", "gender": "Masculino", "id": "98000008", "lastName": "Bravo Bravo", "mobile": "3154575845"}, {"address": "Calle 12 No 12-12 Popayan", "email": "fer@hotmail.com", "firstName": "Alberto", "gender": "Masculino", "id": "98000009", "lastName": "Mendez Bravo", "mobile": "3154575845"}, {"address": "Calle 12 No 12-12 Popayan", "email": "fer@hotmail.com", "firstName": "Alexander", "gender": "Masculino", "id": "98000010", "lastName": "Ponce Yepes", "mobile": "3154575845"}, {"address": "Calle 32 No 8-20 Popayan", "email": "yazmin@gmail.com", "firstName": "Yazmin", "gender": "Femenino", "id": "98000011", "lastName": "Gutierrez", "mobile": "3233408378"}]
```

Cliente creado con el POST